NUMERICAL MODELING OF A SOLID OXIDE FUEL CELL FOR USE IN REAL-TIME SIMULATION AND CYBER-PHYSICAL SYSTEMS

A Thesis
Presented to
The Academic Faculty

by

Jesus Emmanuel Arias

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Mechanical Engineering

Georgia Institute of Technology
May 2019

NUMERICAL MODELING OF A SOLID OXIDE FUEL CELL FOR USE IN REAL-TIME SIMULATION AND CYBER-PHYSICAL SYSTEMS

Approved by:

Dr. Comas L. Haynes, Co-Advisor
Aerospace, Transportation, and Advanced Systems
Georgia Tech Research Institute

Dr. Aklilu T. G. Giorges
Aerospace, Transportation, and Advanced Systems
Georgia Tech Research Institute

Dr. S. Mostafa Ghiaasiaan, Co-Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. David Tucker
Advanced Fuel Cell Division
National Energy Technology Laboratory

Dr. Yogendra K. Joshi
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: April 12, 2019

Towards a better world for my siblings, Lizette and Cesar

ACKNOWLEDGEMENTS

Throughout this journey I have several people to thank for giving me direction. Of course the first person I need to thank is my advisor, Dr. Comas Haynes, for whom I have worked under for several years now. Thanks for giving me the opportunity all those years ago. Despite not doing any of my homework in your fuel cells class, I think I turned out alright, besides the homework was optional. Likewise to Dr. Aklilu T. G. Giorges, thank you for being patient with me despite all the friendly arguing and back and forth, and taking the time to teach me a thing or two. I'm sure you'll still have lots of comments on this thesis, but unfortunately as of the time of writing these acknowledgements, there's no more time left so this will simply have to do. Likewise I must give a thanks to Dr. David Tucker at NETL for letting me run his power plant a week after showing up in Morgantown, which was quite the experience. Nothing I've ever done has been quite as fascinating as working in that Starship Enterprise-esque control center or as difficult as not pushing the giant red button on the control panel. Luckily, nothing went wrong or blew up during my tenure. Another quick thanks to Dr. Dan Maloney for giving me a place to stay while in Morgantown, and likewise to the Hyper Gang for helping me with getting the code to work and also for operating Hyper for all those overly long experiments. Last but certainly not least, to my advisors Dr. Ghiaasiaan and Dr. Joshi for all your support and flexibility in supporting this effort. Finally to close out, I'd like to tell my siblings that I love them more than anything in the world. I dedicate all my work to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

A	Area
α	Charge transfer coefficient
β	Implicit/Explicit weighting factor
c	Specific heat capacity
D	Diffusion coefficient
d_{pore}	Pore Diameter
δ_k	Layer thickness
E_{act}	Activation energy
ε	Porosity
ϵ	Relative error
η	Electrochemical loss
F	Faraday's constant
γ	Pre-exponential factor
ΔG°	Gibbs free energy of formation
H	Convection coefficient
ΔH	Change in enthalpy
I	Current
i_0	Exchange current density
i	Local current density
k	Thermal conductivity
M	Molar mass
$\dot{m}$	Mass flow rate

n_{elec}	Number of electrons transferred per reaction
Δn	Change in molar mass
n	Number of nodes
P	Pressure
P	Channel perimeter
$\dot{Q}$	Heat generation
q'''	Volumetric heat generation per node
ρ_k	Temperature dependent resistivity
ρ	Density
R_u	Universal gas constant
T	Temperature
Δt	Discretized time step
τ	Tortuosity
U	Flow velocity
V	Voltage
ν	Diffusion volume
Δx	Discretized length between nodes
x_i	Mole fraction

Subscripts:

an	Anode
ca	Cathode
c	Cross section
g	Gas

∞	Far field value
K	Knudsen
n	Selected node
p	Constant pressure
SR	Steam reformation
s	Solid
tol	Tolerance
TPB	Triple phase boundary
v	Constant volume
WGS	Water gas shift

SUMMARY

Cyber-physical systems provide a mechanism with which to investigate the physical phenomena and behavior of traditionally cost-prohibitive or otherwise fragile equipment. For the National Energy Technology Laboratory (NETL), this approach resulted in the Hybrid Performance (Hyper) facility which features a gas turbine-SOFC hybrid cycle utilizing real turbomachinery and a simulated SOFC stack. This allows for the investigation of combined cycle performance and control strategies, in an exhaustive manner, both without fear of destroying delicate state-of-the-art fuel cells, and with the full accuracy of real-world turbomachinery. Issues arose between the transient response of the SOFC model being limited to a sample time of 80 milliseconds, due to the calculation time of the SOFC model taking on average 40 milliseconds to calculate for a given timestep with spikes in calculation time reaching the 80 millisecond threshold. In order to be able to match the speed of transients from the turbomachinery and likewise better discern transient behavior, it was determined that the SOFC model must be optimized to operate at a sample time of 5 milliseconds. Therefore, it is necessary to optimize the SOFC model in order to decrease the calculation time from around 40 milliseconds, down to at the most 5 milliseconds. To do this, both the electrochemical algorithm and the thermal algorithm used to simulate the physical behavior of the SOFC are investigated to determine where improvements can be made. To this end the rootfinding numerical recipes of the electrochemical algorithm are investigated as the complex electrochemistry requires a highly iterative nested dual convergence loop to resolve the voltage-current relationship, and likewise the temporal discretization of the thermal algorithm is modified for the sake

of higher accuracy and stability. Ultimately the new electrochemical algorithm featuring higher order rootfinding schemes proves to be efficient enough to reach the sub 5 millisecond target, signifying an order of magnitude reduction in calculation time, and when coupled with the new temporal discretization similar calculation time characteristics show that a fully implicit, higher order temporal discretization can also successfully be used if desired. Ultimately this result means that the cyber-physical simulation system can operate at higher sample rates, and resolve transient events at significantly higher resolution and fidelity.

CHAPTER 1. INTRODUCTION

1.1 Cyber-Physical Systems

Cyber-physical systems (CPS) are becoming more attractive due to the subsequent opportunity to test traditionally cost-prohibitive equipment through a wide range of implementations and conditions. Cyber-physical systems, as defined by the National Science Foundation in their NSF 19-553 program solicitation, provide this capability through the integration of physical hardware to computationally modeled software components coupled through a system of sensors and actuators to mimic physical behavior. This can be thought of as an extension to the existing hardware-in-the-loop approach but expanded upon. If hardware-in-the-loop (HILS) consists of simulating stimuli on a single piece of hardware, then CPS can be thought of as replicating and simulating entire hardware systems through software models and interfacing hardware [1, 2]. Figure 1.1 below provides a visual example of this kind of implementation. The schematic illustrates how a numerical model would be interfaced to physical subsystems through the use of sensors to feed data into computational models, and a transfer system of actuators and controls to replicate the modeled phenomena in a physical manner, to ultimately create a simulated system with more complex and coupled behavior.

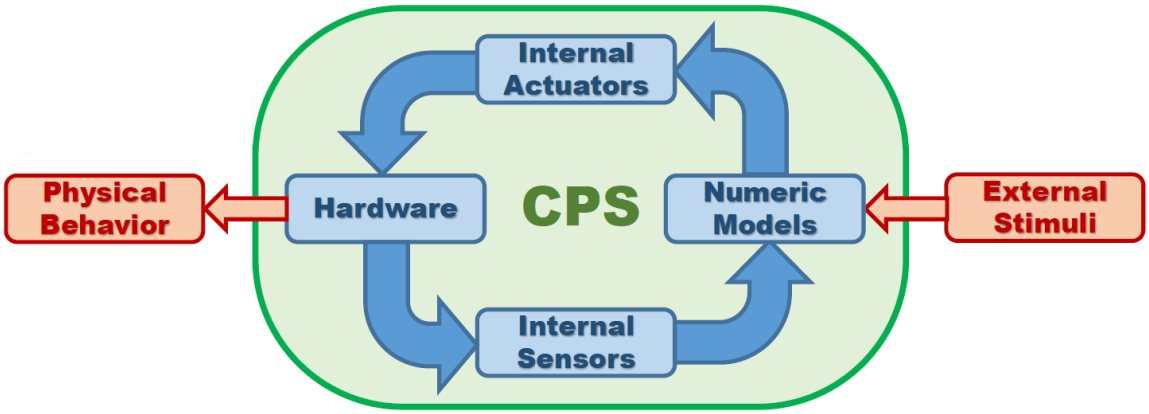

Figure 1.1: Example Schematic of a Cyber-Physical System. Exemplifies the transfer of information that takes place in CPS to simulate real behavior from a balance of computational components and hardware [2].

Real world examples of CPS and HILS range the gamut from application in the automobile industry for testing suspension components [3], autonomous electronic control systems [4], to evaluating the performance of lithium ion batteries for use in hybrid vehicles [5, 6]. As an example, in Hebin [3] a HILS system is used to test an air suspension system, and the subsequent air spring pressure regulation system, necessary for controlling the damping behavior for actual roadway use. To this aim, a simulated road is used to randomly assign load to the air suspension system, and subsequently the response from the air suspension system is used to tune the control scheme of the air suspension system to achieve a particular response, allowing for less real-world development and testing, and also making testing these components safer. Naturally, challenges to CPS include the integrated system response time difference arising from the physical system dynamics, response times of sensors, the computational model and computational time, the control systems, and overall systems interactions [7]. Overall, however, HILS and CPS provide an excellent approach for the development of novel technology with the ability to test systems

that are normally cost prohibitive or simply not developmentally mature enough for feasible real world prototyping and testing.

1.2 Hybrids and Hyper

As will be seen in this thesis, cyber-physical simulation allows for the investigation of solid oxide fuel cell/gas turbine (abbreviated hereon as SOFC/GT) coupling in a safer and more exhaustive manner. The question may arise as to what is so important about a gas turbine/SOFC hybrid cycle that it would warrant investigation. First the concept of an SOFC/gas turbine is explained. Figure 1.2 illustrates a typical open gas turbine Brayton cycle that is typically used for power generation. Below that is Figure 1.3 that shows a simplified SOFC/gas turbine hybrid system, where the combustor is primarily replaced with an SOFC stack which is used to electrochemically oxidize the fuel followed by residual combustion of unutilized fuel. Gaseous products are subsequently expanded in the turbine. Nominally, a standard Brayton cycle gas turbine system has a relatively low electric efficiency of around 20% to 30% even at a large power output as seen in Figure 1.4 [8], whereas a SOFC/GT hybrid cycle has a theoretical electrical efficiency projected to be as high as 60% by 2020 [9], all the way up to 80% electrical efficiency [8, 10].

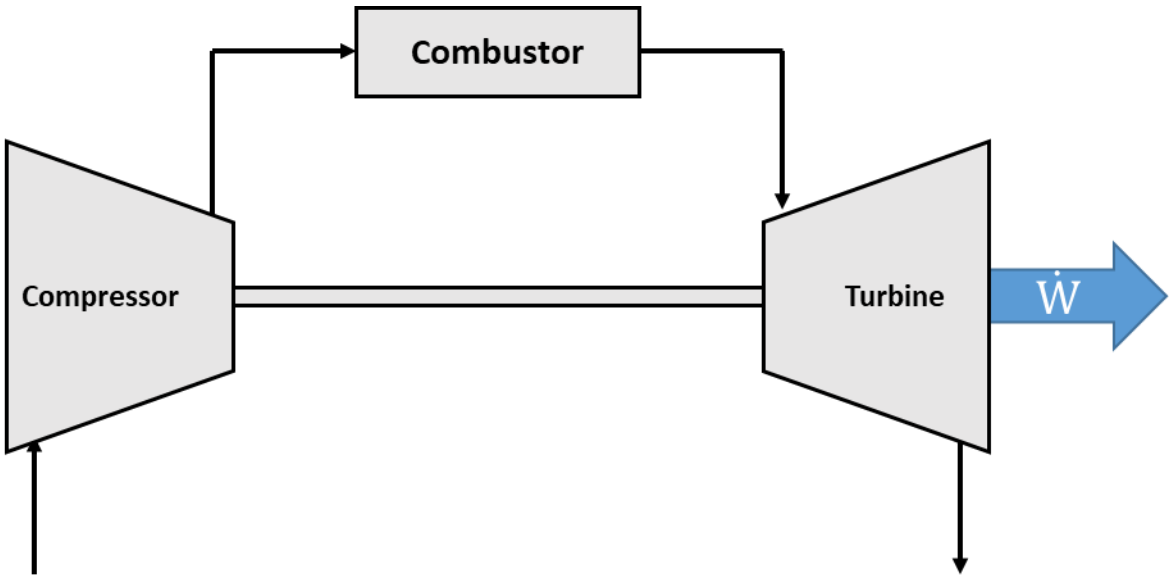

Figure 1.2: Schematic of generic gas turbine Brayton power cycle.

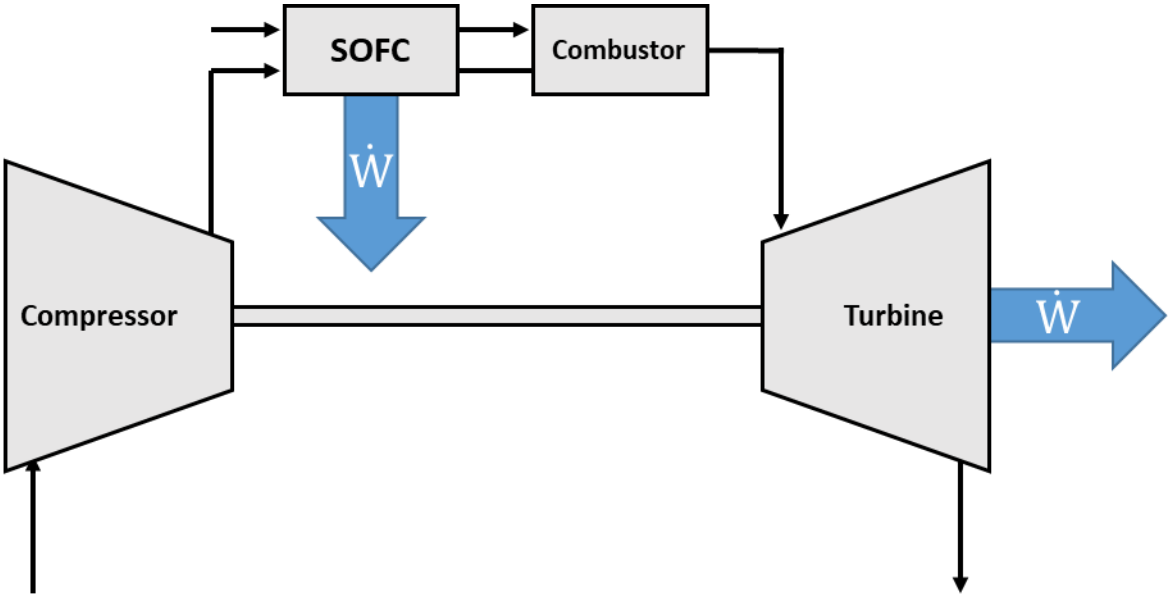

Figure 1.3: Schematic of generic SOFC/GT Hybrid Cycle. Key difference is SOFC stack being placed for catalytic conversion of fuel and oxidant streams, with a combustor only for the combustion of residual fuel.

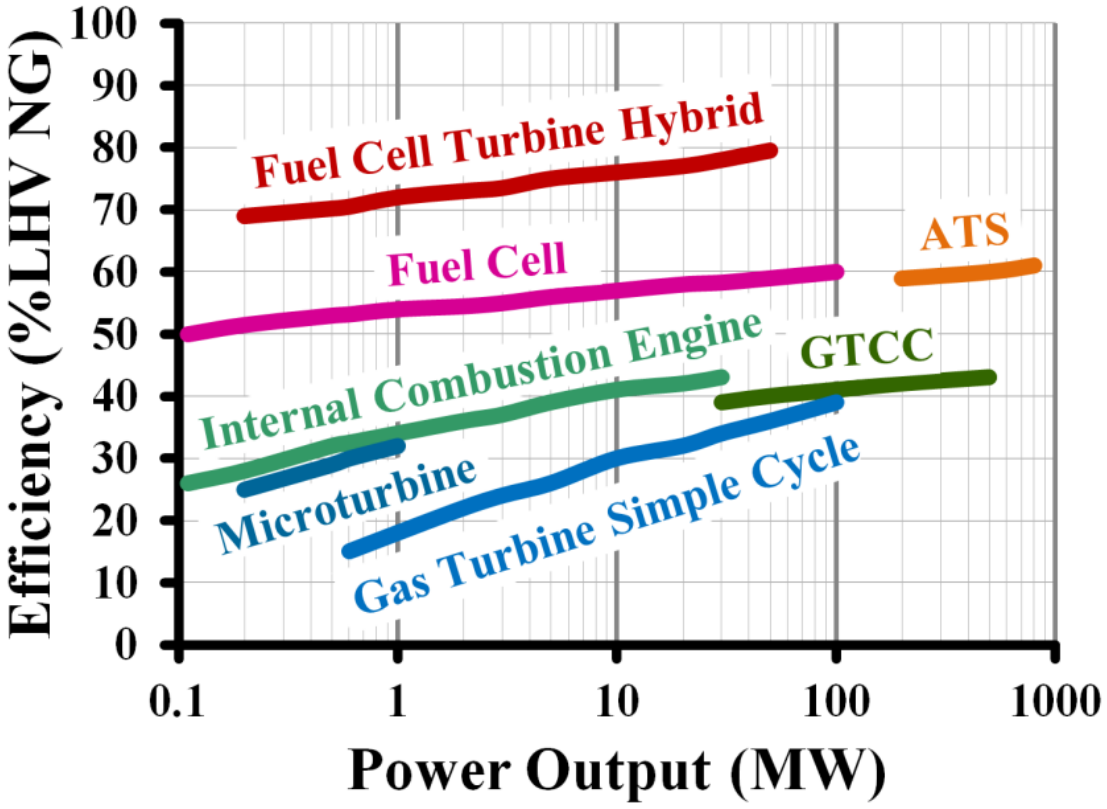

Figure 1.4: Efficiency plot for various energy output paradigms. Exemplifies the drastic difference in efficiency possible for various power generation technologies at different output ranges [11].

Achieving these gains in efficiency, however, is not trivial. In the case of SOFCs, the fragility of a planar fuel cell geometry when subjected to elevated pressures and substantial temperature gradients, combined with their extremely high cost at this phase of development, necessitate the cyber-physical hardware-in-the-loop approach. However, the complexity involved in the computational modeling of such complex systems, necessary to provide accurate results, leads to a difficult trade off regarding how much accuracy can realistically be retained when meeting the time constraints necessary for accurate transient simulation. Resolving this difficulty was the focus of the present thesis.

The National Energy Technology Laboratory (NETL) has developed the Hybrid Performance (Hyper) facility to investigate the performance characteristics and control strategies of a gas turbine-SOFC hybrid cycle featuring physical turbomachinery and a simulated SOFC system. An in-depth schematic of the system is provided as Figure 1.5 and Figure 1.6. The behavior of the SOFC is simulated through interface hardware that both replicates the calculated response from the SOFC model, along with the physical effects of having an additional volume present between the compressor and turbine as illustrated by the yellow blocks in Figure 1.6. These interfacing components are specifically: a fuel valve, FV-432, that adds the appropriate amount of fuel to the gas stream corresponding to the heat generated by the electrochemical reaction of the SOFC and subsequent residual combustion of unused fuel; and an air plenum, V-304, that replicates the volume that would be created by placing an SOFC stack into the gas stream between the compressor and turbine. Additionally, the "Real-Time SOFC System Model", the key area of investigation for this study, receives both live input data from the turbomachinery specifically the flowrate, temperature, and pressure values of the air stream, and additional simulated inputs such as fuel cell load and fuel composition data. This SOFC model component is indicated by the arrow in Figure 1.5. Figure 1.7 features photographs of the actual Hyper facility, illustrating the scale of the system along with some of the specific components.

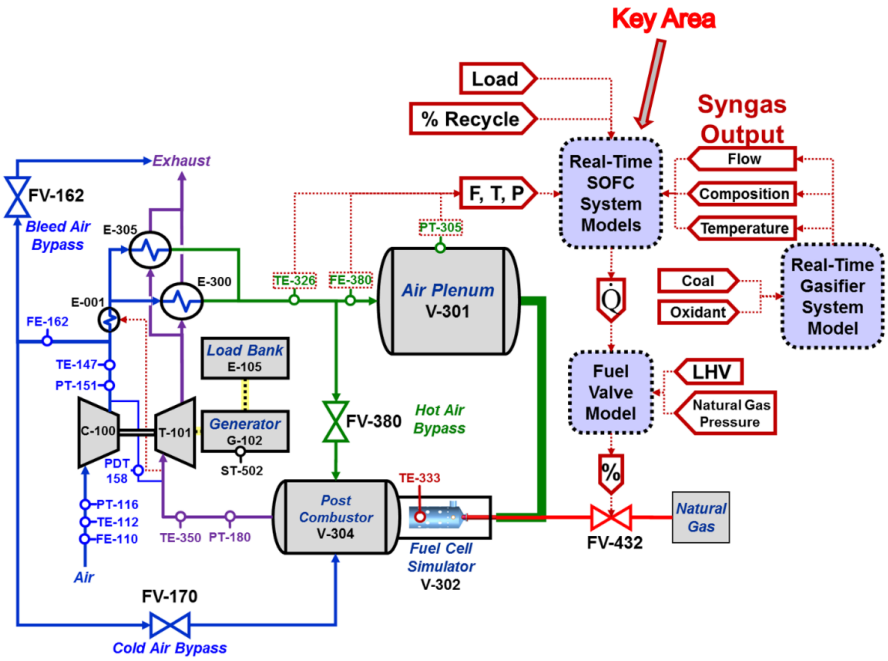

Figure 1.5: Schematic of Hyper Cyber-Physical System with labeled components [1].

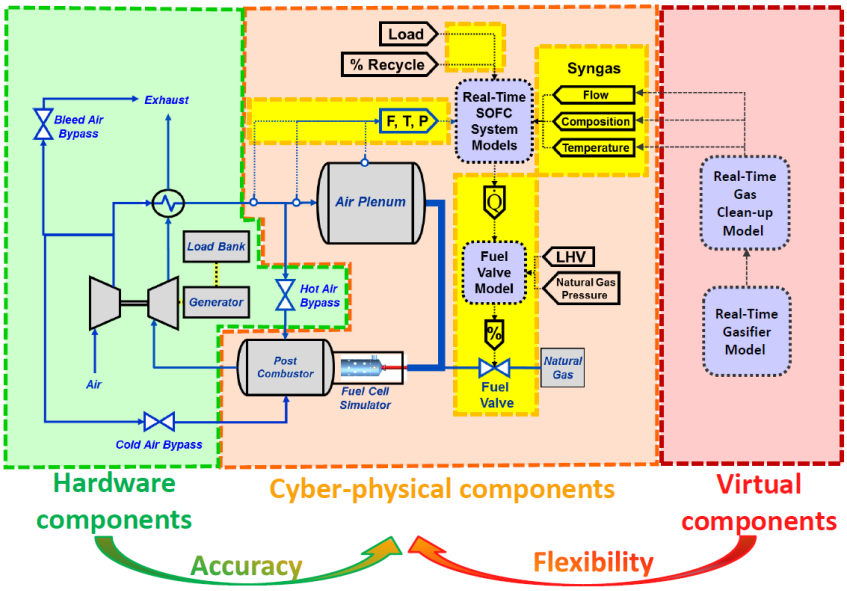

Figure 1.6: Schematic of Hyper Cyber-Physical System highlighting the individual components based on hardware classification with fully virtual components in red, the cyber-physical SOFC in yellow, and the physical gas turbine system and load bank in green. The yellow fields indicate interfacing and middleware [1].

Figure 1.7: Onsite photographs of the Hyper Facility. The leftmost photo shows the machinery and the scale in its entirety. The top right photo is a close up of the air plenum that simulates the volume of the SOFC. The bottom right photo shows the actual turbomachinery used in Hyper, a modified Garrett series 85 auxiliary power unit (APU) rated at 120 kW [12].

1.3 SOFC Simulation Work

The Real-Time SOFC system model used in the Hyper facility is used to calculate several operational parameters within the cell along with the thermal effluent such as solid and gaseous temperature fields, cell voltage, local current density, and chemical compositions of the gas streams as pertinent examples. This is done in order to inform further planar SOFC design and operation. Figure 1.8 illustrates the input/output relationship of the SOFC model. This illustrates how the SOFC model interacts and responds to real conditions, simultaneously taking in live input data to perform its calculations and outputting a thermal effluent value that affects the turbomachinery.

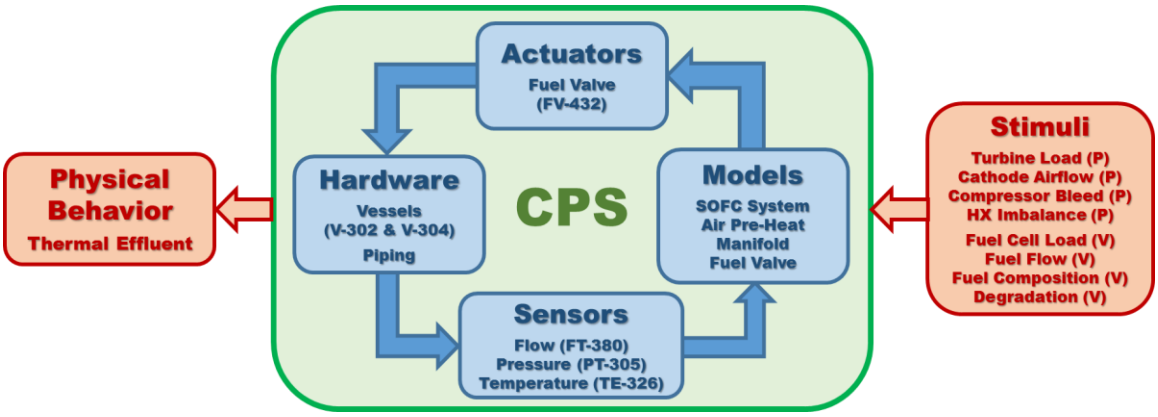

Figure 1.8: Schematic of CPS loop for simulation and replication of SOFC subsystem at Hyper [2].

In the case of NETL's Hybrid Performance (Hyper) facility, issues arose from differing time scales in the response between the physical hardware of the gas turbine system and the simulated solid oxide fuel cell (SOFC) subsystem, particularly concerning the longer computational time of around 40-80 milliseconds. As stated by Tucker et al. [2]:

> "computational predictions (e.g. electrochemical dynamics given load variations inclusive of mass transfer effects) and experimental measurements (e.g. turbomachinery and flow) inform that pivotal responses must be captured at timescales as small as 5 milliseconds."

This study presents two proposed solutions involving the following: 1) optimization of the existing SOFC model; 2) using alternative numerical methods to the current Crank-Nicolson solution scheme. All these measures are proposed in order to decrease simulation time to 5 milliseconds (down from 40 milliseconds) for a 5 millisecond time step, with 5 milliseconds being the operating speed of the gas turbine control mechanisms (specifically fuel valves), likewise representing a decrease in the calculation time by an order of

magnitude, and with the overall aim to better match the two subsystems' transient responses. The ultimate goal is equipping Hyper with the ability to resolve transient behavior in the smallest relevant time scale for use in characterizing said transient behavior for Hyper under a variety of possible real-world scenarios.

CHAPTER 2. DESCRIPTION OF EXISTING SOFC CODE

2.1 Background

A plethora of mathematical models exist for Solid Oxide Fuel Cells of all sorts of varying complexity ranging from 0-dimensional bulk modeling schemes to full 3D. The key constraint for the dimensionality of an SOFC simulation has to do primarily with its proposed application, with lower dimensional schemes typically being used for control applications and higher dimension schemes being used for the actual design and development of SOFC components [13, 14]. An illustration of a typical planar solid oxide fuel cell is presented as Figure 2.1 below.

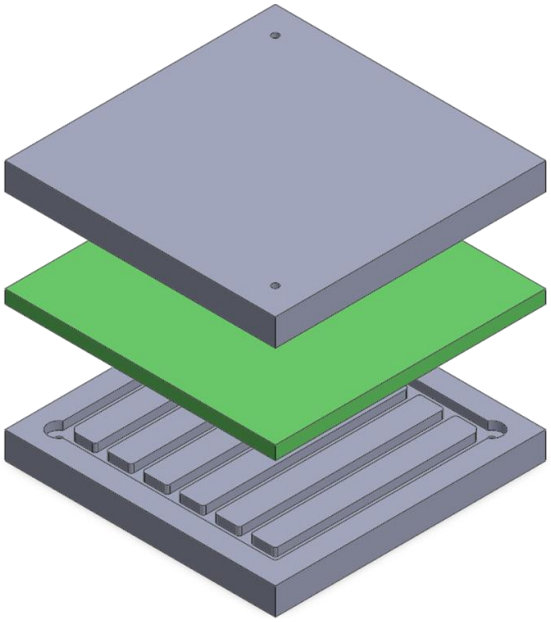

Figure 2.1: Illustration of planar solid oxide fuel cell. The objects in gray are the interconnects which are used to provide reactants to the electrochemically active surface in green. The interconnects also allow the stacking of individual cells into a fuel cell stack.

Primary reasons for the use of lower order models (0D and 1D) for control applications have to do with the orders of reduced calculation time for lower dimensional systems. However, dimensionality is not the only factor at play in the calculation time of a particular SOFC model. Significant computational resources can be expended in the calculation of the electrochemical relationships as the equations become more rigorous to include more effects or the problem becomes less constrained (i.e. non-isothermal conditions, variable local current density, variable voltage) [15].

As defined in Smith's dissertation [11], the original Hyper SOFC computational effort began with a zero dimensional bulk model created by Liese [16]. This allowed for real-time calculation on the relatively less powerful hardware that Hyper used at the time, an AtlasPC by Woodward Industrial Controls [8], which was later changed to use a standalone dSPACE DS1006 board for numerical calculation, and which remained in use through the end of this study. This model allowed for a bulk heat generation to be calculated for a given set of condition and had the added flexibility to calculate the cell voltage for a set current load on the cell, along with other features such as bulk gas stream composition changes.

Hughes in his dissertation [17] later expanded that model to 1D in order to be able to accurately capture the nonlinear spatial behavior that beforehand had to be assumed to linearly change between inlet and exit. It was surmised that in fact, these nonlinearities would directly affect the simulated performance of the fuel cell and as such had to be implemented in at least one dimension along the flow path [2]. Reasons for this include, the balance between cell voltage and local current density for a given cell load. In particular, the local current density could have dramatic effects on the overall cell voltage

and subsequently all other pertinent variables and ultimately the heat generated by the cell [2].

The more recent issues with the Hyper dSPACE platform and the SOFC model arise from the limited computational power of the platform and the highly iterative nature of the SOFC model. In essence, the dSPACE DS1006 board could not run the current implementation of the code any faster, not only establishing a hard limit to the maximum temporal resolution achievable by the platform, but also limiting what additional physics could be incorporated into the real-time SOFC model.

2.2 Current Derivation

The original SOFC code was provided by NETL as developed by Hughes and models the operation of a co-flow SOFC with an electroactive area of 20 cm × 20 cm [12, 17]. The model features a one-dimensional, finite difference discretization of the cell using 20 nodes along the direction of a single channel of flow of length 20 cm as illustrated in Figure 2.2 and Figure 2.4. For specificity Figure 2.3 details the geometry of the SOFC channel and the electroactive components known as the PEN (positive-electrolyte-negative assembly). The first of these figures shows the solid profile for a single channel consisting of the two interconnect components in gray, along with the electrochemically active components (anode, cathode, and electrolyte) in green, as they would be considered in a lumped 1D approach for the simulation of a single channel. The hollow channel region for the oxidant air stream would then be considered the gaseous flow region, providing the species necessary for reaction along with the site for convective heat transfer from the solid region. All the solid components are lumped together, along with the remaining regions

aside from the oxidant stream channel, as illustrated in Figure 2.4 to create the "Solid" region that forms one conservation equation, and then the oxidant stream is considered as a bulk flow field for the "Gas" region. Additionally, a multiplicative effect is used to simulate the results of the multiple flow channels present inside of the SOFC as illustrated in Figure 2.5.

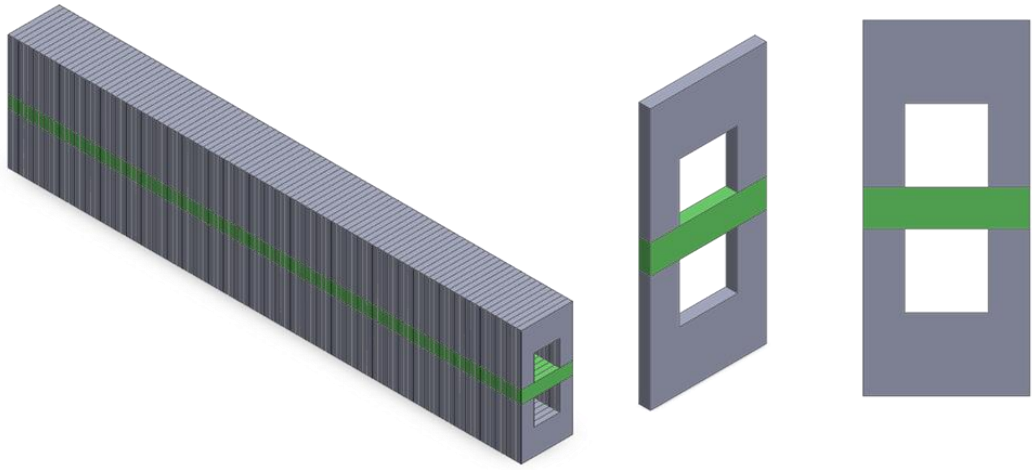

Figure 2.2: Nodal Discretization of a single SOFC channel. Gray and green components form the solid region of the cell, and the hollow regions in the gray interconnect form the flow path for the fuel stream and oxidant stream.

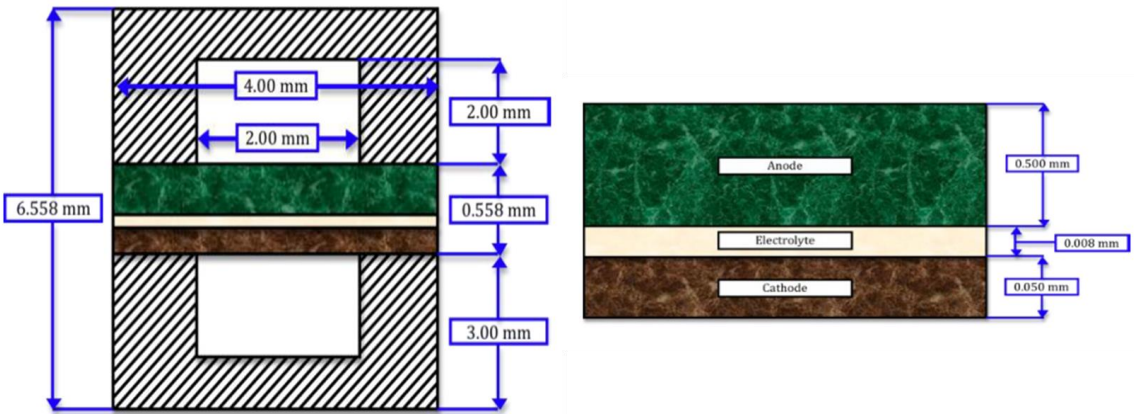

Figure 2.3: Geometric parameters of the SOFC channel and PEN [17].

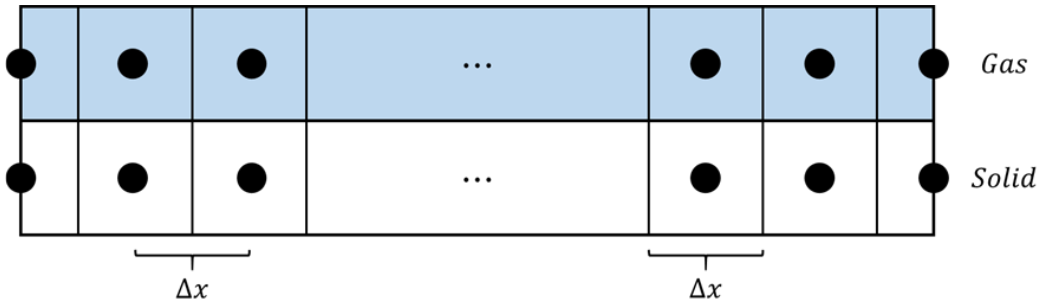

Figure 2.4: Schematic of bulk nodalization of gaseous and solid phases along a channel. The ends of the regions feature half-volumes done to resolve the boundary edge effects of the SOFC.

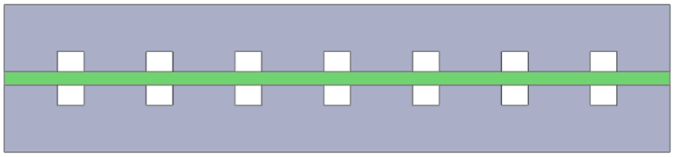

Figure 2.5: Cross-section of SOFC illustrating multiple channels. The results from the calculation of a single SOFC flow channel is multiplied by the number of channels to estimate the overall performance of a single planar cell.

As stated prior, SOFC models of this form are easiest to classify based on what degrees of freedom are present, or otherwise what constraints are made for the sake of ease of calculation. The code derived by Hughes, features an isopotential, one-dimensional cell, where all other relevant parameters are allowed to vary according to external stimuli. Under this model, the fuel cell is considered to have a constant voltage throughout the cell, however the voltage itself is able to vary based on fuel cell load demand, temperature, and species concentration. The derivation for such is provided in Hughes' dissertation [17] and provided below in the following section for thoroughness. Likewise, the thermal characteristics are modeled using a one-dimensional, finite difference discretization of the cell also outlined in the sections that follow. The values of the operational parameters of the model are presented in Appendix A under Table A.1. The operational inputs of the

model are the cathode inlet parameters of temperature, pressure, and mass flow rate, the total number of cells, current demand, fuel flow, and fuel composition.

2.2.1 Electrochemical Algorithm

The electrochemical algorithm for the solid oxide fuel cell revolves mainly around the calculation of the cell voltage as all subsequent calculations require this term. Likewise, the cell voltage is calculated from the load across the cell and electrochemical losses related to operational conditions as defined by Equation (2.1).

$$\mathrm{V}_{cell} = \mathrm{V}(i) = \mathrm{V}_{Nernst} - \eta_{dif} - \eta_{act} - \eta_{ohm} \tag{2.1}$$

This formulation has the cell voltage being calculated by four terms, with V_{Nernst} being the Nernst Potential (thermodynamic maximum reversible voltage), and three different modes of electrochemical losses, being diffusion polarization η_{dif}, activation polarization η_{act}, and ohmic losses η_{ohm}. These loss mechanism pertain to the following phenomena: 1) Diffusion polarization stemming from concentration gradients throughout the cell; 2) Activation polarization arising from the existence of a non-zero activation energy required to drive the electrochemical reaction and subsequent current throughout the cell; 3) Ohmic losses resulting from the flow of current through an internally resistive media and the subsequent decrease in voltage arising from such. The actual functional forms of these terms are defined in a plethora of SOFC modeling literature but some more-so standardized interpretations are presented below [12, 16-20].

The Nernst Potential of the cell is determined from Equation (2.2) and functionally stems from the change in Gibbs free energy of reaction $\Delta G^{\circ}_{H_2O}$ along with the main

environmental factors of temperature T and species partial pressures P_x. Likewise, R is the universal gas constant and F is Faraday's constant.

$$V_{Nernst} = -\frac{\Delta G^{\circ}{}_{H_2O}}{2F} + \frac{RT}{2F} ln\left(\frac{P_{H_2} P_{O_2}^{1/2}}{P_{H_2O}}\right) \tag{2.2}$$

These form the maximum theoretical voltage that could be produced by the cell, and such Nernst potential is used as the limiting factor in fuel cell calculations.

The activation polarization as stated arises from the non-zero amount of energy required to initially drive the electrochemical reactions in the fuel cell. This corresponds to the first region in Figure 2.6.

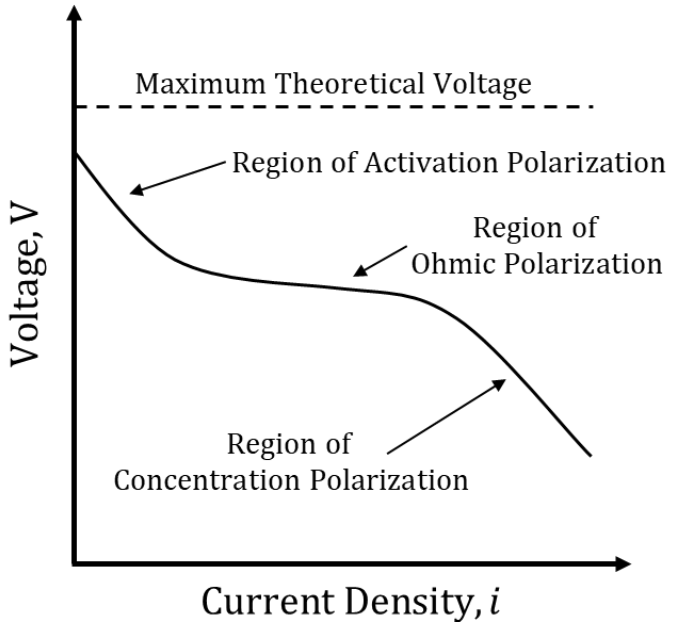

Figure 2.6: Illustration of a typical polarization curve

The derivation of the activation polarization term, η_{act}, used in the SOFC code stems from the Butler-Volmer equation presented below as Equation (2.3).

$$i = i_o \left(\exp\left(\frac{\alpha n_{elec} F \eta_{act}}{RT}\right) - \exp\left(\frac{-(1-\alpha) n_{elec} F \eta_{act}}{RT}\right)\right) \tag{2.3}$$

A simplified and explicit alternative form from Noren and Hoffman [21] is used by Hughes [12] in order to reduce the calculation from an implicit transcendental form, into a closed expression given as Equation (2.4).

$$\eta_{act} = \left(\frac{RT}{\alpha n_{elec} F}\right) \sinh^{-1}\left(\frac{i}{2i_o}\right) \tag{2.4}$$

In this formulation the activation polarization can be directly calculated using the local current density i and the exchange current density i_0. The other miscellaneous terms, α and n_{elec}, are the charge transfer coefficient and the number of electrons transferred per reaction respectively. The exchange current density, i_o, must be derived for both the anode and cathode, as each of these sites of electrochemical reaction add to the total activation polarization present within the cell, and these expressions Hughes adopts from Li [22] and are presented below as Equations (2.5) and (2.6) with p_{amb} being the ambient pressure, p_x as species partial pressure, and E_{act} being the activation energy required for each.

$$i_{o.an} = \gamma_{an} \left(\frac{p_{H_2}}{p_{amb}}\right)\left(\frac{p_{H_2O}}{p_{amb}}\right) \exp\left(-\frac{E_{act,an}}{RT}\right) \tag{2.5}$$

$$i_{o,ca} = \gamma_{ca} \left(\frac{p_{O_2}}{p_{amb}}\right)^{0.25} \exp\left(-\frac{E_{act,ca}}{RT}\right) \tag{2.6}$$

The ohmic polarization is calculated from the losses occurring from current flowing through a resistive media, and the SOFC code accounts for the resistance of the anode, cathode, and electrolyte assembly, R_{PEN}, using Equation (2.7) with R_{PEN} itself being calculated using Equation (2.8).

$$\eta_{ohm} = i\left(A_{eff} R_{PEN}\right) \tag{2.7}$$

$$R_{PEN} = \sum_{k=an,ca,ele} \left[\frac{\rho_k \delta_k}{A_k}\right] \tag{2.8}$$

Here A_{eff} is the effective cross-sectional area of the PEN, which is comprised of the electroactive anode, cathode, and electrode. Likewise, ρ_k, δ_k, and A_k are the temperature dependent resistivity, the thickness, and cross-sectional area of each component respectively.

The diffusion polarization term however varies, having several functional derivations depending on the complexity of the diffusion model being employed at the triple phase boundary (TPB). Previously Smith [11] used a more simplified model using only a scalar coefficient for the diffusion effects and consequently the molar ratios at the triple phase boundary. Hughes, however, incorporates the effects of Fick's law, Knudsen diffusion, and ordinary binary diffusion using the Fuller-Schettler-Giddings method. The base equation for the diffusion polarization is defined by Equation (2.9).

$$\eta_{dif} = \frac{RT}{2F}\left(\ln\left(\frac{x_{H_2,bulk}}{x_{H_2,TPB}} \cdot \frac{x_{H_2O,TPB}}{x_{H_2O,bulk}}\right) + \frac{1}{2}\ln\left(\frac{x_{O_2,bulk}}{x_{O_2,TPB}}\right)\right) \tag{2.9}$$

From Hughes, the mole fractions of the reactants at the triple phase boundaries are calculated using Equations (2.10 – 2.12), essentially as a deviation from the bulk mole fractions $x_{x,bulk}$ for each species.

$$x_{H_2,TPB} = x_{H_2,bulk} - \frac{RT}{2F} \cdot \frac{i\delta_{an}}{p_{an} D_{an,eff}} \tag{2.10}$$

$$x_{H_2O,TPB} = x_{H_2O,bulk} + \frac{RT}{2F} \cdot \frac{i\delta_{an}}{p_{an} D_{an,eff}} \tag{2.11}$$

$$x_{O_2,TPB} = 1 + \left(x_{O_2,bulk} - 1\right) exp\left(\frac{RT}{4F} \cdot \frac{i\delta_{ca}}{p_{ca} D_{ca,eff}}\right) \tag{2.12}$$

The effective diffusion coefficients for the anode and cathode side, $D_{an,eff}$ and $D_{ca,eff}$, respectively are calculated using Equations (2.13) and (2.14).

$$D_{an,eff} = \left(\frac{p_{H_2O}}{p_{an}}\right) D_{H_2,eff} + \left(\frac{p_{H_2}}{p_{an}}\right) D_{H_2O,eff} \tag{2.13}$$

$$D_{ca,eff} = D_{O_2,eff} \tag{2.14}$$

Likewise, the effective diffusion rates for the pertinent electrochemical species are calculated by combining the effects of Binary and Knudsen diffusion using Equations (2.15 – 2.17) with ε and τ being the porosity and tortuosity of the medium respectively.

$$D_{H_2,eff} = \frac{\varepsilon}{\tau}\left(\frac{1}{D_{H_2,H_2O}} + \frac{1}{D_{K,H_2}}\right)^{-1} \tag{2.15}$$

$$D_{H_2O,eff} = \frac{\varepsilon}{\tau}\left(\frac{1}{D_{H_2,H_2O}} + \frac{1}{D_{K,H_2O}}\right)^{-1} \tag{2.16}$$

$$D_{O_2,eff} = \frac{\varepsilon}{\tau}\left(\frac{1}{D_{O_2,N_2}} + \frac{1}{D_{K,O_2}}\right)^{-1} \tag{2.17}$$

Binary diffusion, stems from Fick's law and its coefficients are calculated with the Fuller-Schettler-Giddings correlation below as Equations (2.18) and (2.19), with the effective molar masses, M_{H_2,H_2O} and M_{O_2,N_2}, being defined as Equations (2.20) and (2.21). Here ν is the diffusion volume for each species.

$$D_{H_2,H_2O} = \frac{0.00143 \cdot T^{1.75}}{pM_{H_2,H_2O}^{1/2}\left[(\sum\nu)_{H_2}^{1/3} + (\sum\nu)_{H_2O}^{1/3}\right]} \tag{2.18}$$

$$D_{O_2,N_2} = \frac{0.00143 \cdot T^{1.75}}{pM_{O_2,N_2}^{1/2}\left[(\sum\nu)_{O_2}^{1/3} + (\sum\nu)_{N_2}^{1/3}\right]} \tag{2.19}$$

$$M_{H_2,H_2O} = 2\left(\frac{1}{M_{H_2}} + \frac{1}{M_{H_2O}}\right)^{-1} \tag{2.20}$$

$$M_{O_2,N_2} = 2\left(\frac{1}{M_{O_2}} + \frac{1}{M_{N_2}}\right)^{-1} \tag{2.21}$$

Knudsen diffusion depends on the pore diameter d_{pore} of the porous substrate of the electrochemically active areas [23] and its coefficients are calculated using Equations (2.22 – 2.24) for each relevant species.

$$D_{K,H_2} = 48.5 \cdot d_{pore} \left(\frac{T_{PEN}}{M_{H_2}} \right)^{1/2} \tag{2.22}$$

$$D_{K,H_2O} = 48.5 \cdot d_{pore} \left(\frac{T_{PEN}}{M_{H_2O}} \right)^{1/2} \tag{2.23}$$

$$D_{K,O_2} = 48.5 \cdot d_{pore} \left(\frac{T_{PEN}}{M_{O_2}} \right)^{1/2} \tag{2.24}$$

Due to the degree of freedom allowed in the calculation of the cell voltage in the fuel cell particularly, and the allowance of a variable local current density for a given load condition, a nested approach must be used. This is done to satisfy the two conditions that constrain the voltage-current relationship given by Equations (2.25) and (2.26). The constraints that these equations apply are: for (2.25) that the total cell current generated for a particular voltage guess, $\mathrm{I}(\mathrm{V}_{guess})$, matches the set current load, I_{load}; and for (2.26) that voltage produced for a particular local current density guess, $\mathrm{V}(i_n)$, matches the set iteration voltage, V_{guess}. An illustration of this process is provided below as Figure 2.7.

$$\mathrm{I}(\mathrm{V}_{guess}) - \mathrm{I}_{load} = 0 \tag{2.25}$$

$$\mathrm{V}(i_n) - \mathrm{V}_{guess} = 0 \tag{2.26}$$

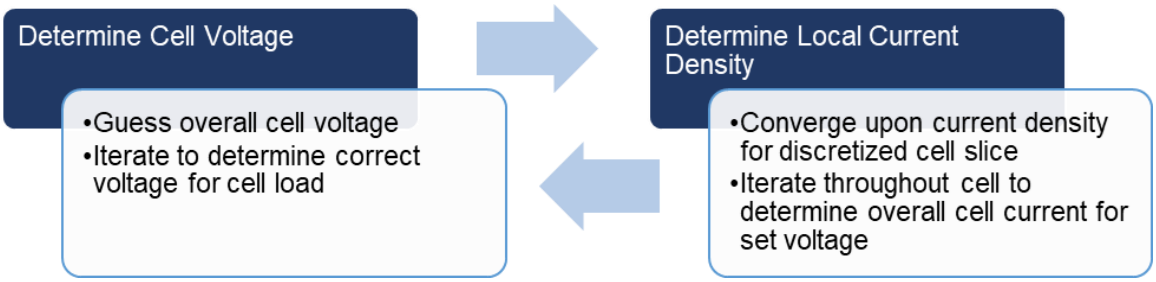

Figure 2.7: Flowchart of iterative process for determining Voltage-Current relationship for fuel cell. Both use rootfinding recipes to determine values and subsequently must be solved simultaneously for a given timestep.

These two equations are solved concurrently using rootfinding recipes. For Equation (2.25) the algorithm used is the bisection method. This method is a bracketed scheme that uses a window to solve for the root of a function as illustrated by Figure 2.8 and codified by Equations (2.27) and (2.28).

$$V_c = \frac{V_a + V_b}{2} \tag{2.27}$$

$$\text{If } \operatorname{sgn}(I_{total}(V_a) - I_{load}) * \operatorname{sgn}(I_{total}(V_c) - I_{load}) < 0 \tag{2.28}$$

$$\text{Then } V_b = V_c, \text{ otherwise } V_a = V_c$$

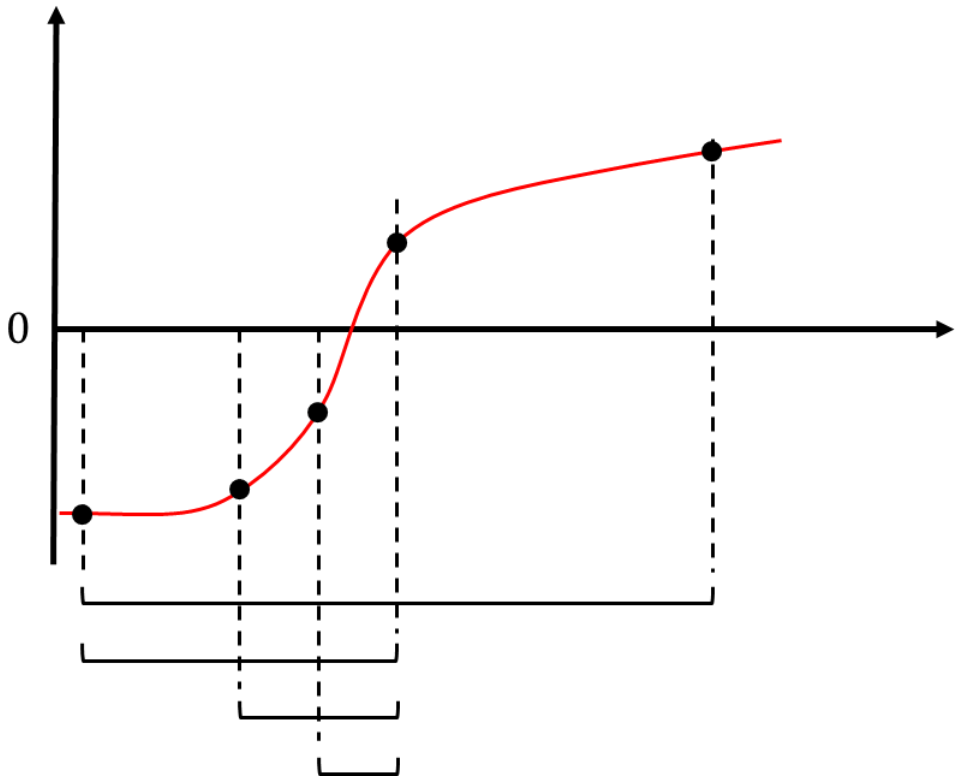

Figure 2.8: Illustration of bisection method solving for the root of an arbitrary function. The bracketing window decreases in size by half while maintaining one positive and one negative value at the ends of the bracket.

For Equation (2.26) the algorithm used is the false position method as defined by Equations (2.29)-(2.31).

$$i_{c,n} = i_{a,n} - \frac{f_{a,n} * \left(i_{b,n} - i_{a,n}\right)}{f_{b,n} - f_{a,n}} \tag{2.29}$$

$$f_{x,n} = \mathrm{V}\left(i_{x,n}\right) - \mathrm{V}_{\mathrm{guess}} \tag{2.30}$$

$$\text{If } f_{a,n} * f_{c,n} < 0 \tag{2.31}$$

Then $i_{b,n} = i_{c,n}$, otherwise $i_{a,n} = i_{c,n}$

This specific process is investigated further as one of the main avenues of optimization presented in this study. The reasoning behind the use of these schemes, are that they are both bracketed techniques that are stable and will always find a root, given that one exists. This comes with the caveat of a lower order of convergence which will also be investigated in Chapter 3.1 [24].

2.2.2 Thermal Algorithm

The thermal algorithm uses two separate tridiagonal matrices to define each system of equations, one for the solid phase incorporating all the SOFC components and regions aside from the oxidant stream channel, and the other for the oxidant gas stream (i.e. a two-temperature field model). The solid phase discretization takes into account convective heat transfer occurring between the oxidant stream channel carved into the interconnect and the oxidant gas stream, and a source term arising from heat generation due to the electrochemical reactions present in the fuel cell. The overall discretization of the solid phase is presented as Equation (2.32). The derivation used by Hughes employs the Crank-Nicolson scheme to discretize the temporal dimension. This technique uses a blend of implicit and explicit values delineated by the factor β set to 0.74, with $\beta=0$ corresponding to a fully explicit scheme, and $\beta=1$ corresponding to a fully implicit scheme. The subscripts of variable T correspond to the temperature at that node location as presented in Figure 2.9 and the subscripts correspond to the particular time step with n+1 corresponding to the next time step.

$$\beta\left[\frac{kA_c}{\Delta x^2}(T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}) + hA_g(T_\infty - T_i^{n+1})\right]$$

$$+(1-\beta)\left[\frac{kA_c}{\Delta x^2}(T_{i+1}^{n} - 2T_i^{n} + T_{i-1}^{n})\right] + q'''A_c(\Delta x) \qquad (2.32)$$

$$= \rho c_p \Delta x\left[\frac{T_i^{n+1} - T_i^n}{\Delta t}\right]$$

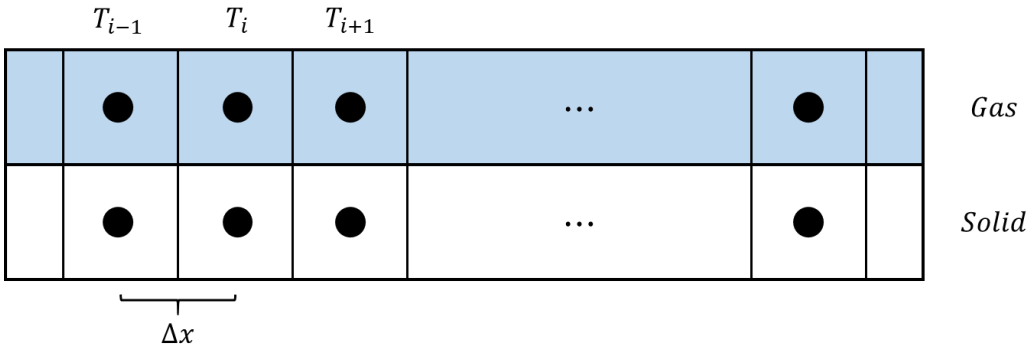

Figure 2.9: Discretization of PDEs with illustration of indexing for discrete temperature values.

The heat generation term is derived from all the reactions along with the thermodynamic heat of reaction and electrochemical losses occurring in the SOFC as indicated in Equation (2.33). The terms included are the byproduct heat from the production of H_2O from the main electrochemical reaction as seen in Equation (2.34), along with the heat produced from the water gas shift reaction as Equation (2.35) and from steam reformation as Equation (2.36).

$$q''' = HG_{cell} + HG_{WGS} + HG_{SR} \qquad (2.33)$$

$$HG_{cell} = i\left(\frac{-\Delta H_{H_2O,Formation}}{2F} - V\right) \tag{2.34}$$

$$HG_{WGS} = -\Delta n_{CO}\Delta H_{WGS} \tag{2.35}$$

$$HG_{SR} = -\Delta n_{CH_4}\Delta H_{SR} \tag{2.36}$$

The system of equations is then solved for the solid and gaseous regions concurrently through the use of the tridiagonal matrix algorithm as developed by Thomas [25].

2.3 Existing Performance and Issues to Resolve for Real-Time Processing

As alluded to in Ch. 1.3 and informed by the study from Tucker et al. [2], the performance of the SOFC code when executed on the dSPACE platform is on the order of 40 to 80 milliseconds as reported by NETL [2]. To reiterate, this calculation time is not sufficient, being too slow to resolve all relevant transient events during coupled operation of the cyber-physical fuel cell and the gas turbine system, examples of which include simulated electrochemical dynamics and experimental measurement resulting from changes in turbomachinery operation and flow, which can have timescales as small as 5 milliseconds [2]. This conclusion directly drives our push for reducing the calculation time down to lower than 5 milliseconds.

Figure 2.10 illustrates the relationship between the calculation time of the model and the sample time of the cyber-physical system, and furthermore how that translates to resolving transient responses in the cyber-physical system along with the actual time step Δt of the SOFC model. Essentially, the cyber-physical system can only respond to stimuli as quickly as its computational components can calculate said response. This functionally

means that the model time step Δt is limited by and can only be as small as the longest model calculation time. Ergo, in order to resolve all transient features of interest at higher temporal resolution, i.e. using a smaller Δt for the model calculations, then the algorithm must converge quickly enough to produce the results at the next time step, $t = t + \Delta t$, before the next sampling time, which incidentally is the next time step.

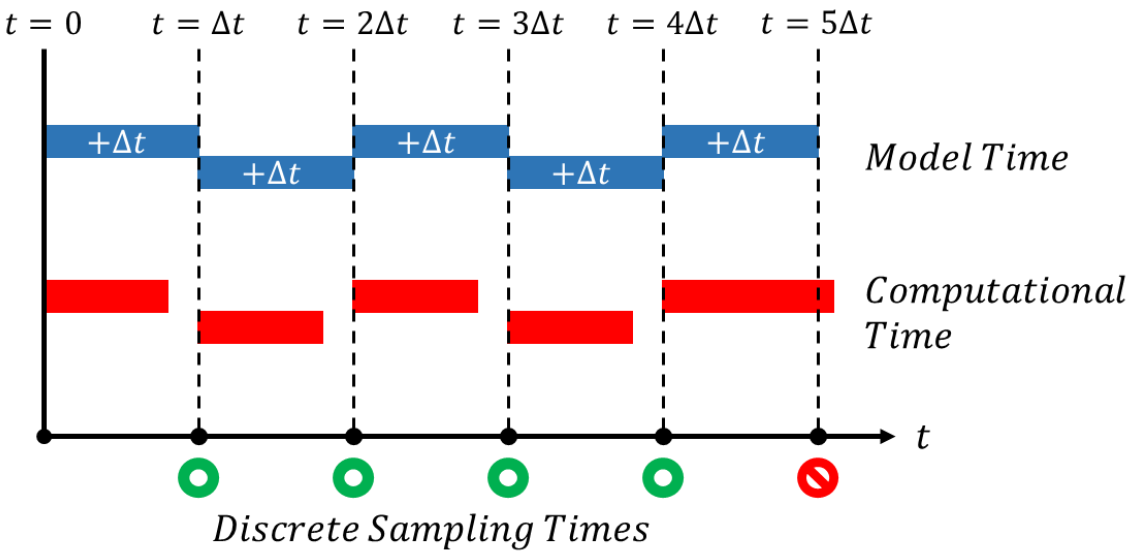

Figure 2.10: Timing diagram relating computational time, to the time step being represented in the SOFC model and the discrete times at which values are sampled by the dSPACE platform. Illustrates how the model time step must match the discrete sampling time and that likewise the computation for a given time step cannot take longer than these values.

To achieve this crucial goal, the first step was to determine what processes in the SOFC model were taking up the most computational time. This was done by examining the underlying MATLAB code that contains the SOFC model itself, since for operation in dSPACE it must first be implemented into a Simulink model which is then compiled into a dSPACE executable. The SOFC MATLAB script (known specifically as "Combined_Heating") was executed using the MATLAB Profiler. This program generates

reports of the times each function takes to execute, and the total time spent in each function. Figure 2.11 shows the results of the profile.

Function Name	Calls	**Total Time**	Self Time*	Total Time Plot (dark band = self time)
Automator_Tavg	1	319.960 s	0.059 s	
ExecCode_Combined_Heating_Tavg	2	319.519 s	0.053 s	
combined_heating	2	285.295 s	0.137 s	
echeatgenv4	51	276.170 s	8.957 s	
CURRENTDENSITYFUNC	265094	263.758 s	14.413 s	
VOLTERROR	3467415	249.345 s	249.345 s	
Write2Excel	1	33.641 s	0.015 s	

Figure 2.11: Code execution results for a sample calculation until steady state. As indicated by the dark blue bar, algorithm spends most calculation time in VOLTERROR which is called upon for the solution of Equation (2.26).

Immediately apparent from these results is that the majority of the calculating time is spent in the electrochemical algorithm, contained in the function “echeatgenv4” and subfunctions “CURRENTDENSITYFUNC” and “VOLTERROR”. Specifically, the self-time of VOLTERROR (i.e. the amount of time spent inside that particular function), makes up the vast majority of the calculation time with a disproportionate number of calls to it. Since this corresponds directly to the dual rootfinding scheme, it becomes quickly apparent that some sort of constraint on expedience exists in the electrochemical algorithm and work on optimizing and streamlining must be implemented in these sets of functions.

CHAPTER 3. MODIFICATIONS TO SOFC ALGORITHM

3.1 Optimizing the Electrochemical Algorithm

As the electrochemical sub-algorithm appears to affect the overall calculation time most significantly, it is key to reduce this computational burden. Two primary solution paths were investigated for the optimization of the electrochemical algorithm; the first being the use of new higher order rootfinders, and the second being the use of modified convergence tolerances. The aim of this two-pronged approach was to alleviate both bottlenecks and alleviate impedance to rapid convergence.

3.1.1 Modification of Convergence Parameters

The nested rootfinding algorithm used a set relative tolerance to determine adequate convergence when solving for the cell voltage and the current density for each node. For the voltage scheme, the code execution stops when Equation (3.1) is satisfied.

$$|\mathrm{I}_{total} - \mathrm{I}_{load}| < \mathrm{I}_{tol}, \qquad \mathrm{I}_{tol} = (1 \times 10^{-3})\mathrm{I}_{load} \tag{3.1}$$

For resolving the current density, the code was halted upon the condition defined by Equation (3.2).

$$|i_n - i_{n,previous}| < i_{n,tol},$$

$$i_{n,tol} = (1 \times 10^{-3}) \frac{i_{limit}}{\mathrm{A}_{elec} \cdot n} \tag{3.2}$$

Here i_{limit} is the limiting current density, A_{elec} is the electroactive area, and n is the total number of nodes. Together these conditions served to ensure that overall the voltage algorithm converged in step with the overall cell load.

Due to the exceedingly large amount of time spent in the electrochemical algorithm, these conditions were investigated to ensure the relative tolerance was appropriate for the problem. It was decided to ultimately redefine the convergence tolerance to ensure a relative tolerance of 1e-3. This was decided upon the reasoning that 99.9% relative accuracy (i.e. an allowed variance of 0.1% between iterations) would be enough to resolve the voltage and current density, without significantly slowing down the code in one specific area. It appeared as though, due to a particularly tight tolerance for determining the current density at each node, the inner convergence loop was where the primary amount of time is spent on calculations. The tolerance for the inner loop was changed to Equation 3.3 to resolve this.

$$|i_n - i_{n,previous}| < i_{n,tol},$$

$$i_{n,tol} = (1 \times 10^{-3}) \frac{i_{limit}}{n} \tag{3.3}$$

The primary reason for removing the A_{elec} term was to avoid the double counting of the per-unit-area term (i.e. local current density divided by local electroactive area)

which significantly tightened the tolerance. Additionally the limiting current density, which was defined from reactant mass transport principles, was already in units of [A/cm^2] as needed for the direct comparison with the current density tolerance and therefore another division by A_{elec} further divided the current density term and erroneously made the units [A/cm^4]. By redefining and loosening the overly constricting convergence tolerance for the inner loop, which is executed nominally twenty times for each iteration of the outer voltage algorithm, significant decreases in calculation time were possible due to the multiplicative effect of making the inner most algorithm more efficient.

3.1.2 Rootfinder Investigation

With the convergence tolerance for the nested algorithm redefined, further investigation was done to determine the convergence properties of the nested rootfinding algorithm itself. Originally, the nested electrochemical algorithm from Figure 2.7 was performed using the bisection method for the outer voltage loop and the false position method for the inner current density loop. The inner loop was designed to use the false position method, specifically because it has a higher order of convergence than the bisection method [24, 26]. This naturally led to investigating the convergence properties of various rootfinding algorithms. Figure 3.1 illustrates the convergence properties of the rootfinding methods chosen for investigation, with those being the bisection method, false position method, secant method, and Newton method. Here the simple function $0 = \ln x$ is solved for its root by each method.

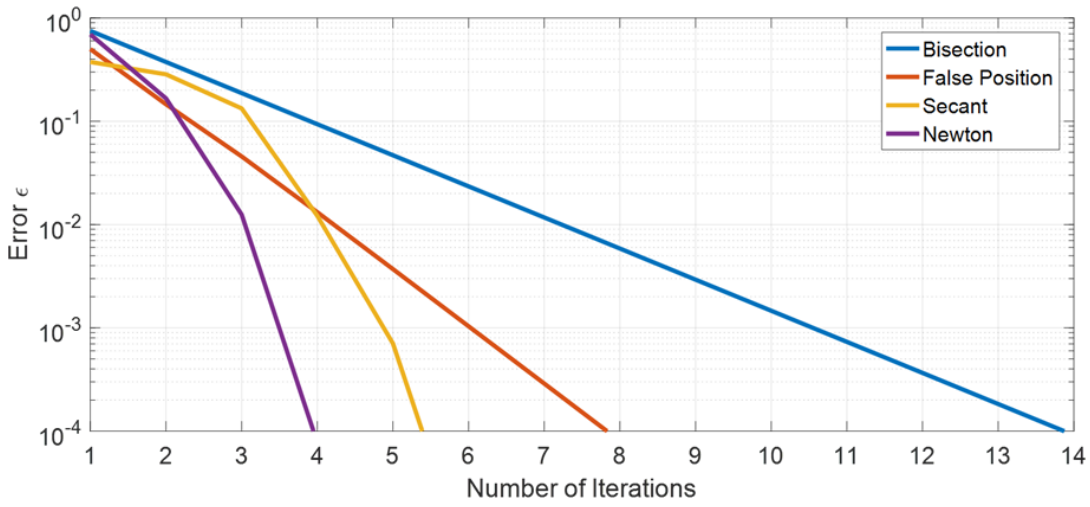

Figure 3.1: Plot of Relative Error per Number of Iterations for different rootfinding methods. Study performed on representative problem $0 = \ln x$. Illustrates the potential for optimization by using higher order rootfinding schemes. However, faster convergence is not guaranteed.

Immediately apparent from the plot is that the error term for each method varies and has the potential to decrease dramatically faster. From Burden's Numerical Analysis [26] this results from the order of convergence from each method. The bisection method is a bracketed method and has the lowest order of convergence at 1 which, as evident from the error plot, means that the bisection method can only decrease the error at a fixed linear rate. However, this also means that the solver cannot solve any slower than the fixed rate of the method, and likewise will decrease the error term no matter the function, since the only information the method needs is the sign of the values for the function at each root estimate. This functionality is important as will be explained in the following section. The false position method retains the bracketing scheme of the bisection method, however, it uses an estimation of the slope to determine the next bracket. This allows the scheme to converge more quickly, however it is still a linear scheme and the order of the error will likewise decrease in a linear fashion similar to bisection but at a faster rate. The secant

method differs from the previous two rootfinding recipes, in that it is an unbounded scheme. In this method, once again an estimation of the slope of the function being investigated is used to iteratively solver for a root. For this method however, the method iterates from the previous root estimate and directly continues from that point only. This leads to the higher order of convergence of 1.62 [26] and as shown in Figure 3.1. It allows the method to essentially home in on a root once close enough and reduce the error term at an increasingly faster rate with each subsequent iteration. Lastly, if the function in question has a differentiable form, then the Newton method can be employed which features a quadratic order of convergence [26] once in the neighborhood of a root and likewise is the fastest a rootfinding algorithm can converge upon a root.

3.1.3 Replacing Electrochemical Algorithm: Implementation of Higher Order Rootfinders

The implementation of higher order rootfinders into the existing electrochemical algorithms presents special challenges stemming from the structure of the code itself and the way certain variables are calculated. Specifically, higher order rootfinders require there to be a functional form with which to evaluate the function being investigated with the current value of the root, or in other words have the form presented below as Equation (3.4) be an explicit function that can be directly evaluable.

$$y = f(x) \tag{3.4}$$

This is acutely problematic when considering Equation (3.5).

$$\mathrm{I}_{\mathrm{generated}} = \mathrm{I}(\mathrm{V}_{\mathrm{guess}}) \tag{3.5}$$

As stated prior, this function does not have a closed form and cannot be explicitly determined. This is the entire reason that the voltage and current density must be determined simultaneously using a nested rootfinding approach. Previously the original method for the determination of cell voltage used the bisection method which simply needs the sign of $\mathrm{I}(\mathrm{V}_{guess}) - \mathrm{I}_{load}$ to determine how to modify the bracketing for the next iteration. Essentially this allowed for the algorithm used for Eqn. (2.26), $\mathrm{V}(i_n) - \mathrm{V}_{guess} = 0$, to proceed throughout the cell and terminate as soon as the total cell current was too high, expediting the calculation by breaking the algorithm early. This concept is illustrated in Figure 3.2.

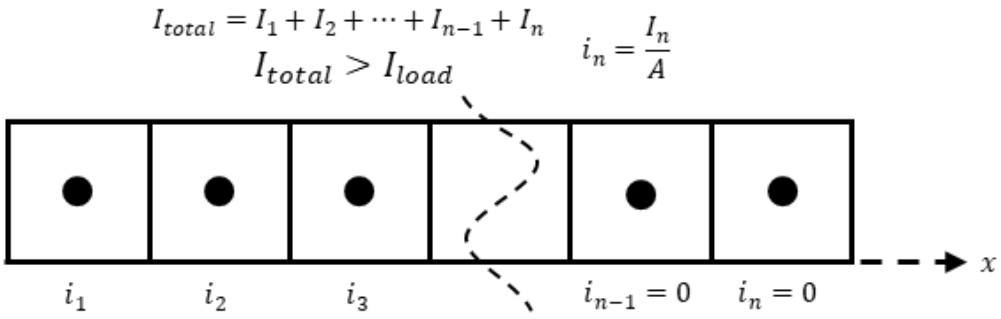

Figure 3.2: Schematic of current density algorithm sweeping throughout the computational fuel cell nodes. Algorithm uses rootfinder to solve equation (2.26) to find local current density which is converted to the amount of current generated in the slice. The slice currents are then added up to determine the total amount of current generated by the cell. If the calculated current becomes higher than the prescribed load the algorithm terminates early.

This functionality not only had to be kept for the sake of accelerated calculation, but also had to be expanded upon to estimate the current for the whole cell as a function of voltage as needed for the use of higher order rootfinders. In order to perform this, a linear

extrapolation is used to estimate the current density throughout the entire cell, by extrapolating the current density from the first node and the node where the total current goes above the set load current. This concept is illustrated in Figure 3.3. As previously expanded on by Arias [7]:

> "This approach is suitable since as the cell voltage converges, likewise the cell current also converges, resulting in the extrapolation error tending to zero. This is due to the calculated current being less likely to surpass the load current until further down the fuel cell as the voltage converges, leading to eventually having the extrapolation going away entirely. In essence, this allows the electrochemical algorithm to be adaptive and only fully resolve the current density for all nodes when the relative error of the voltage algorithm is low and necessitates it."

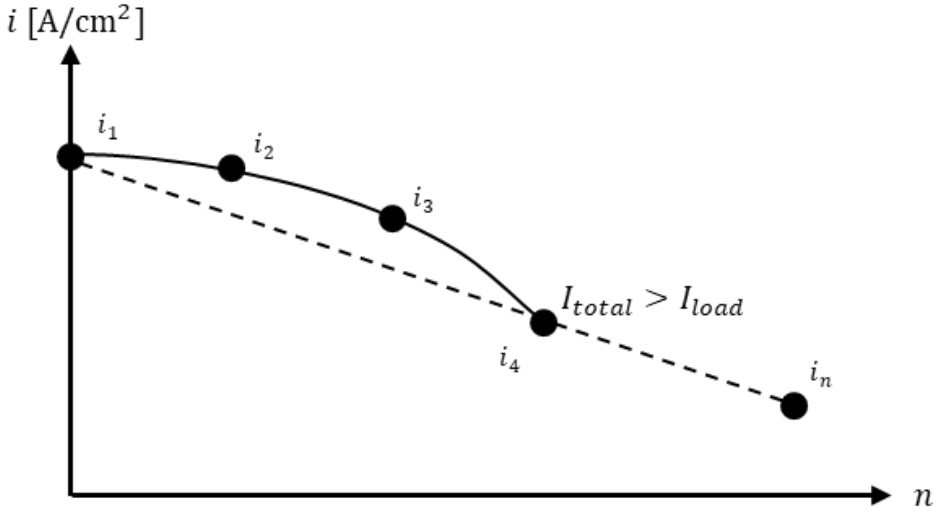

Figure 3.3: Schematic of current density algorithm along with the current density extrapolation scheme. The current density at the end of the cell is extrapolated using the current density at the first node and the current density when the total cell current surpasses the load current.

With the current density estimation scheme defined, the higher order rootfinding scheme can be fully defined and implemented into the voltage calculation scheme. The computational loops are illustrated by Figure 3.4 and Figure 3.5 for calculating the voltage and the local current density respectively. The actual equations for each method are as follows for voltage:

False	If $f_a * f_c < 0$	(3.6)
Position:	Then $\mathrm{V}_b = \mathrm{V}_c$, otherwise $\mathrm{V}_a = \mathrm{V}_c$	
Secant:	$f_a = f_b, \mathrm{V}_a = \mathrm{V}_b$ $f_b = f_c, \mathrm{V}_b = \mathrm{V}_c$	(3.7)
Next Root	$\mathrm{V}_c = \mathrm{V}_b - \frac{f_b * (\mathrm{V}_b - \mathrm{V}_a)}{f_b - f_a}$	(3.8)
	$f_x = \mathrm{I}_{\mathrm{total}}(\mathrm{V}_x) - \mathrm{I}_{\mathrm{target}}$	(3.9)

And as follows for local current density:

False	If $f_{a,n} * f_{c,n} < 0$	(3.10)
Position:	Then $i_{b,n} = i_{c,n}$, otherwise $i_{a,n} = i_{c,n}$	
Secant:	$f_a = f_b, i_{a,n} = i_{b,n}$ $f_b = f_c, i_{b,n} = i_{c,n}$	(3.11)

Next Root $$i_{c,n} = i_{a,n} - \frac{f_{a,n} * \left(i_{b,n} - i_{a,n}\right)}{f_{b,n} - f_{a,n}} \tag{3.12}$$

$$f_{x,n} = \mathrm{V}\left(i_{x,n}\right) - \mathrm{V}_{\mathrm{guess}} \tag{3.13}$$

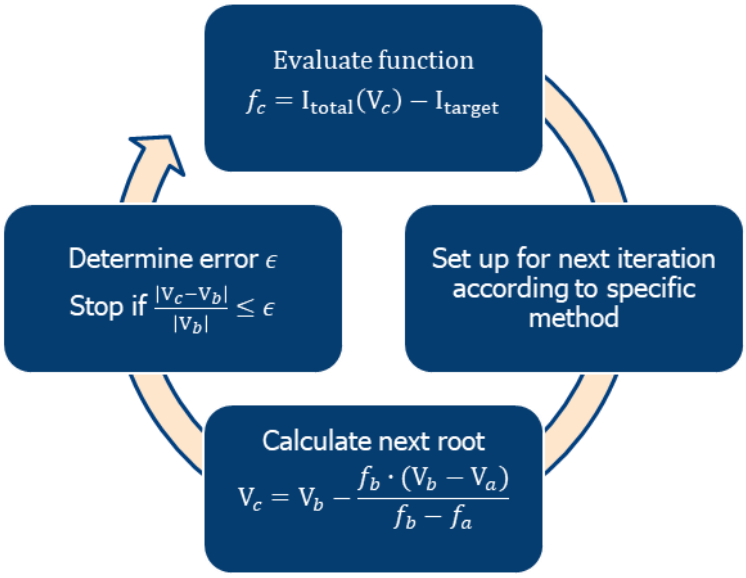

Figure 3.4: Diagram of rootfinding algorithm as implemented in voltage finding scheme. Implements current density estimation scheme and uses the results to allow for the implementation of higher order rootfinders. Cycle continues until a converged cell voltage is reached.

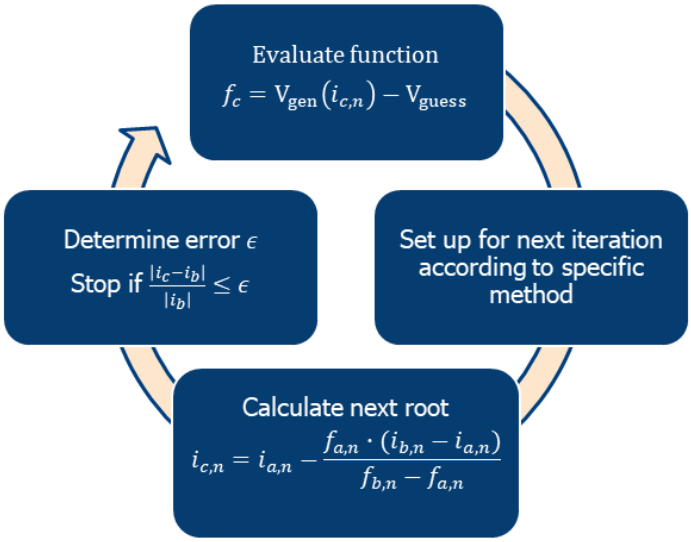

Figure 3.5: Diagram of rootfinding algorithm as implemented in local current density finding scheme. Cycle continues until a converged local current density is reached.

Lastly, with all these numerical recipes defined for each computational loop, three new configurations were developed and implemented as illustrated in Table 3.1 using a relative convergence tolerance of 1e-3, along with a Modified Bisection method for a fourth configuration using the original algorithm but with the new tolerancing also presented in Table 3.1. In Arias [7] the choice was made to exclude the Newton method from consideration as:

> "its derivation for the current density algorithm is extremely tedious and the degree of instability that is inherent in such a numerical recipe was deemed too high for the sharp stimuli that the numerical system will experience, specifically in regard to noisy inputs."

Regardless the other numerical methods are investigated in full moving forward.

Table 3.1: Name of algorithm and the associated numerical method used for solving each equation.

Algorithm name:	**Equation (3.25)** $\mathbf{I}(\mathbf{V}_{guess}) - \mathbf{I}_{load} = \mathbf{0}$	**Equation (3.26)** $\mathbf{V}(i_n) - \mathbf{V}_{guess} = \mathbf{0}$
Modified Bisection	Bisection	False Position
False Position	False Position	False Position
Secant	Secant	False Position
Double Secant	Secant	Secant

3.2 Replacing Thermal Algorithm: What is Crank-Nicolson?

The other main sub-algorithm used to simulate the behavior of the SOFC is a thermal model that considers the heat generated from electrochemistry and simulates thermal transport throughout the fuel cell solid region and oxidant gas stream. Originally the thermal algorithm used the Crank-Nicolson time marching scheme, which uses a trapezoidal approximation of the time derivative akin to Equation (3.14).

$$\frac{\partial T}{\partial t} \approx \frac{\beta T - (1-\beta)T'}{\Delta t} \tag{3.14}$$

The key distinction for Crank-Nicolson is the blending of implicit and explicit methodology. When applied to the governing equation of the SOFC, the result is Eqn. (2.32) as previously defined, and the accompanying numerical stencil is presented as Figure 3.6.

$$\beta\left[\frac{kA_c}{\Delta x^2}(T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}) + hA_g(T_\infty - T_i^{n+1})\right]$$

$$+(1-\beta)\left[\frac{kA_c}{\Delta x^2}(T_{i+1}^{n} - 2T_i^{n} + T_{i-1}^{n})\right] + q'''A_c(\Delta x) \tag{2.32}$$

$$= \rho c_p \Delta x\left[\frac{T_i^{n+1} - T_i^n}{\Delta t}\right]$$

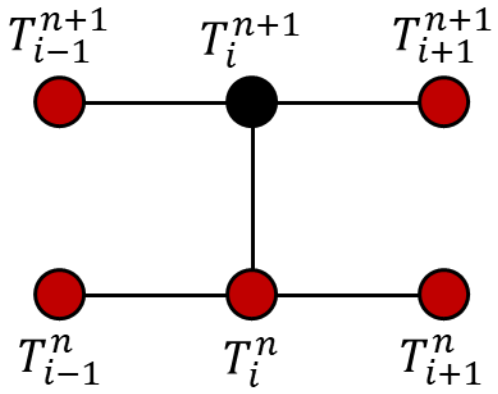

Figure 3.6: Nodal stencil for Crank-Nicolson scheme. Illustrates the nodes used to calculate the value of T_i^{n+1} featuring a blended implicit and explicit formulation.

Features of this derivation include second-order temporal accuracy assuming a β value of 0.5. Likewise, the blend of implicit and explicit formulation, give the method some of the faster calculation speed of explicit methods, and a higher stability envelope from the implicit formulation. However, Crank-Nicolson is not unconditionally stable and for the implementation in Hyper, a β value of 0.74 is utilized meaning a more implicitly weighted scheme which was done specifically for stability reasons [17]. This caveat degenerates the method to first order in time despite the added complexity of the derivation [27]. For this reason, an alternative was proposed for the discretization of the thermal governing equation with the goal of a higher order of accuracy for the time derivative, and possibly an

unconditionally stable formulation. The expectation is that it will likewise result in better convergence of the overall method due to higher accuracy in time.

3.2.1 Three-Point Time Marching Scheme

The major change proposed for the discretization of the time derivative is substituting the trapezoidal time discretization of Crank-Nicolson with a three-point backwards difference approximation for the time derivative. The new approximation for the time derivative is shown below as Equation (3.15). When applied to the governing equation of the SOFC solid phase the resulting derivation is Equation (3.16) and the accompanying stencil for the numerical scheme is illustrated as Figure 3.7.

$$\frac{\partial T}{\partial t} \approx \frac{3T - 4T' + T''}{2\Delta t} \tag{3.15}$$

$$\frac{kA_c}{\Delta x^2}(T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}) + hA_g(T_\infty - T_i^{n+1}) + q'''A_c(\Delta x) = \frac{\rho c_p \Delta x}{\Delta t}\left(\frac{3}{2}T_i^{n+1} - 2T_i^n + \frac{1}{2}T_i^{n-1}\right) \tag{3.16}$$

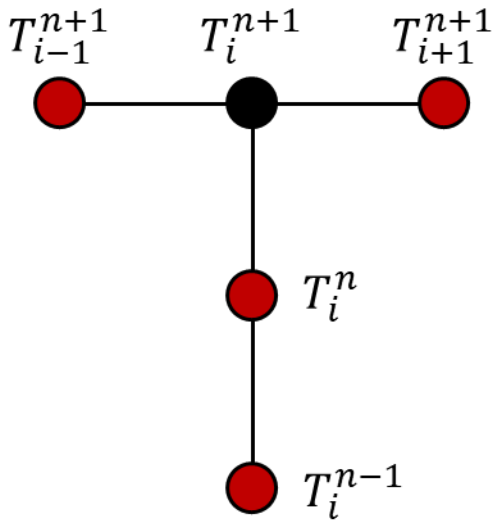

Figure 3.7: Nodal stencil for Three-point Backwards Difference scheme. Illustrates the nodes used to calculate the value of T_i^{n+1} featuring a fully implicit formulation. The key difference of this scheme is the use of two prior time steps to better approximate the time derivative.

This new method was chosen due to its second order accuracy in time [28]. Additionally, it is a fully implicit method that is likewise unconditionally stable. Lastly, due to a better approximation of the time derivative, it is hoped that upon the initialization of a new time step, that the initial values will be closer to those of a converged solution, and likewise the electrochemical algorithm will need to be executed fewer times overall. The main caveat of using this method is the additional memory requirements stemming from the new call to an additional previous time step.

CHAPTER 4. EXPERIMENTAL METHOLOGY AND RESULTS

4.1 Development of Transient Test Cases for Use Offline/MATLAB Testing

In order to be able to test the behavior of all of the newly derived SOFC algorithms, a MATLAB platform was created in order to be able to implement them and test the relative convergence properties of each. This was accompanied with a set of tests to be able to standardize the performance during sharp transients (e.g. severe step changes). This chapter discusses the methodology used along with any and all relevant parameters referenced accordingly.

4.1.1 Input Parameter Investigation

The primary test for characterization of the convergence properties is to induce sharp step changes to the main input parameters to the SOFC code. These inputs are specifically the cell load, the fuel composition flowing into the cell, and the fuel and air relative flow rates and are used to test the fully integrated compiled version of the new algorithms on dSPACE. For the initial development and characterization in MATLAB, however, the primary input parameter that was changed was the current load on the fuel cell as this immediately and explicitly affects the electrochemical algorithm. This preliminary investigation was presented in full in a previous publication by Arias [7] and the results of which are included in this thesis as a critical first step. Below are Figure 4.1 and Figure 4.2 with the first of the results at a baseline steady current load which showed great promise despite being on MATLAB's interpretive code platform. For these tests all

inputs were kept constant, and the results stem from allowing the solution to reach steady state from the initial condition of both gas and solid streams being set to 1000K.

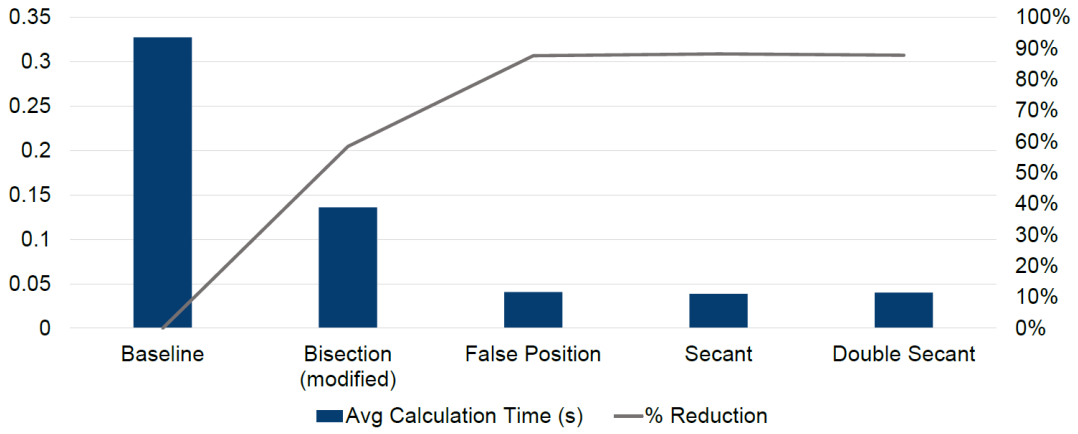

Figure 4.1: Plot of average calculation time in seconds for different rootfinding methods along with the percent reduction in calculation time from baseline. Model parameters are for a load of 250 A, 80% fuel utilization, time steps Δt of 40 ms, and input temperature of 1000 K.

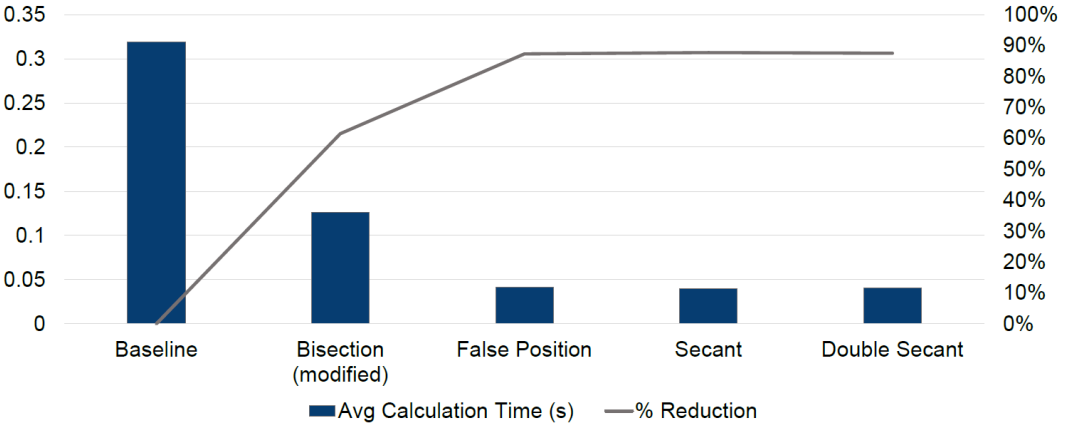

Figure 4.2 Plot of average calculation time in seconds for different rootfinding methods along with the percent reduction in calculation time from baseline. Model parameters are for a load of 350 A, 80% fuel utilization, time steps Δt of 40 ms, and input temperature of 1000 K.

As seen from the results on the MATLAB platform, even just redefining the convergence tolerances reduced the calculation time by nearly 60%. Furthermore, the

electrochemical methods using higher order rootfinders reduced the calculation time as much as 90%. As stated prior, due to the interpretive nature of MATLAB code, the actual calculation time is significantly higher than milliseconds and as such only relative reductions are relevant. However, reductions on this order of magnitude are very significant and give credence to the possible success on Hyper. Before moving on to Hyper however, first the stability with regards to both convergence and of calculation time, are investigated in order to see how sharp stimuli affects each numerical method. The results of this preliminary current load spike test are presented below as Figure 4.3.

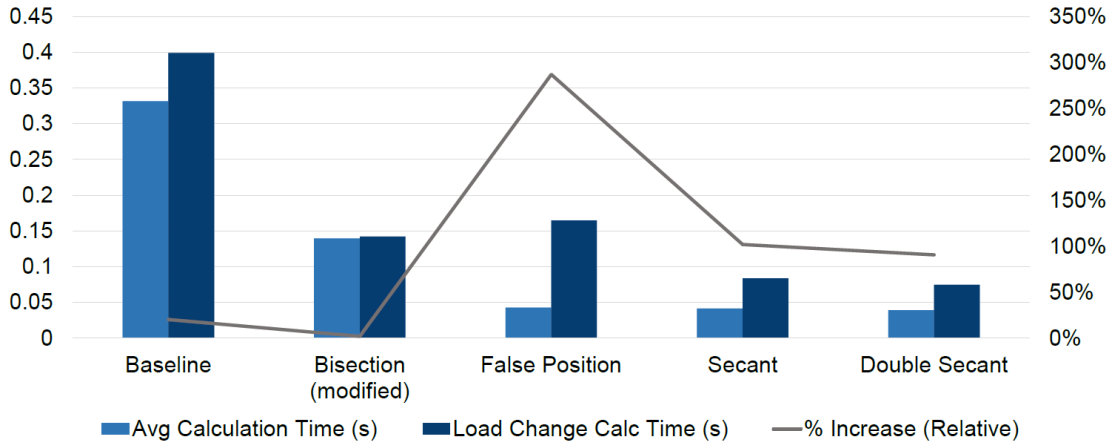

Figure 4.3: Plot of average calculation time and calculation time during a load step change event, both in seconds, and the percent relative increase in calculation time from the average time to the load change time (i.e. resulting spike in calculation time during step change) for each rootfinding scheme. Model parameters are for an input load of 250 A at a 50% fuel utilization that is then increased to 95% fuel utilization resulting in a final load of 450 A.

The results from the load spike on the cell reveal certain particularities and vulnerabilities in each electrochemical method. First of which is that sharp stimuli significantly affect the calculation time of the higher order rootfinders. Secondly, that different schemes are affected differently. For the baseline method, the relative increase

was not very significant, and the same appeared with the modified bisection method that also features almost no increase in calculation time. This is somewhat expected for the fixed linear method. The key difference comes from the false position method, which has the highest relative spike in calculation time and ultimately surpasses even the modified bisection method during the load spike. Better news, however, is that the secant and double secant calculation methods appear to be much less susceptible to sharp transients and likewise still calculate the fastest.

4.2 Results and Discussion: Testing of Code Features

Based on the preliminary results on MATLAB, full implementation on dSPACE was the next milestone. This involved porting the new MATLAB scripts into the functional Simulink project that is then compiled into a dSPACE executable for running on the dSPACE DS1006 processor board. Subsequently, a process of gradual characterization was performed on the platform, first with constant inputs to simply see if the algorithms were stable and to establish relative performance figures. Live inputs from the plant were investigated next to determine stability with noisy inputs. Finally, full coupling to the turbomachinery was performed, with the algorithm both receiving live inputs and also outputting a heat generation term that directly resulted in additional fuel going into the turbine. The initial values for the input fields used are available in Appendix A under Table A.2.

4.2.1 Performance with Constant Inputs

For the initial characterization on dSPACE, the code was first tested purely with load changes starting off at a cell current load of 200 A, increased to 220 A, 240 A, up to

a maximum of 250 A, and ending with a return to 200 A. These sharp stimuli were performed to ensure a strong response from the electrochemical algorithm and subsequently had the longest computational time possible. The first set of results were done at the original time step of 80 milliseconds, done so in order to be able to directly compare to the original algorithm provided by NETL. The following three figures (Figure 4.4 – Figure 4.6) feature and highlight the results of the first batch of testing. Immediately apparent from Figure 4.6 is the stark difference in turnaround time, analogous to the calculation time plus any additional overhead before starting a new iteration, between the different algorithms, as alluded to in the initial results in MATLAB. The key takeaway was the reduction in the original (Original – CN) calculation time of around 40 milliseconds down to below 5 milliseconds for the secant (S – CN) and double secant (DS – CN) methods. In the case of the false position method (FP – CN), an interesting phenomenon was evident in Figure 4.5 where the first calculation after the load trip is accompanied by a corresponding jump in calculation time but then quickly subsides to calculation times similar to secant and double secant. This was somewhat expected as the false position method is a blend of the secant methods' root approximation technique coupled with the bisection methods' bracketing. Overall however, at least for these results using only the Crank-Nicolson time marching scheme, it appeared as if only secant and double secant are capable of sub 5 millisecond calculation.

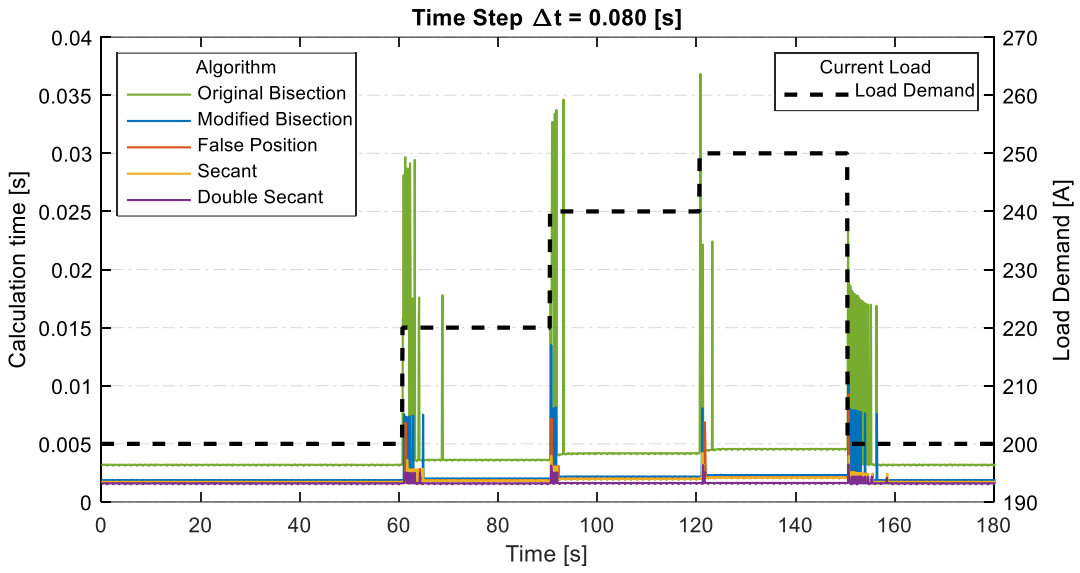

Figure 4.4: Plot of calculation time in seconds for different rootfinding methods during drastic load changes. Using Crank-Nicolson time marching scheme. Simulation running at Δt = 80 ms.

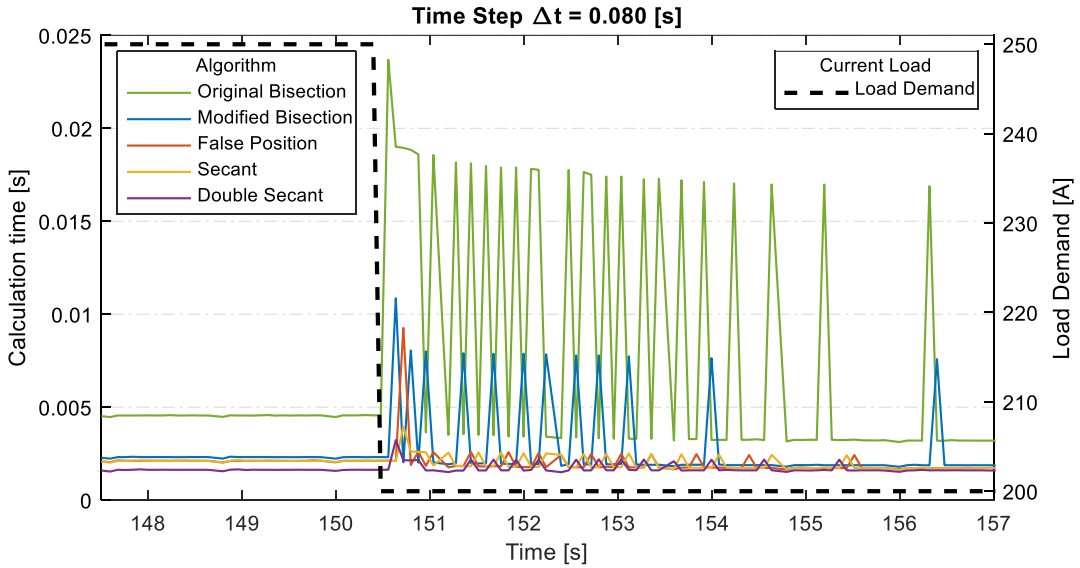

Figure 4.5: Zoomed in plot of calculation time in seconds for different rootfinding methods during drastic load changes. Using Crank-Nicolson time marching scheme. Simulation running at Δt = 80 ms.

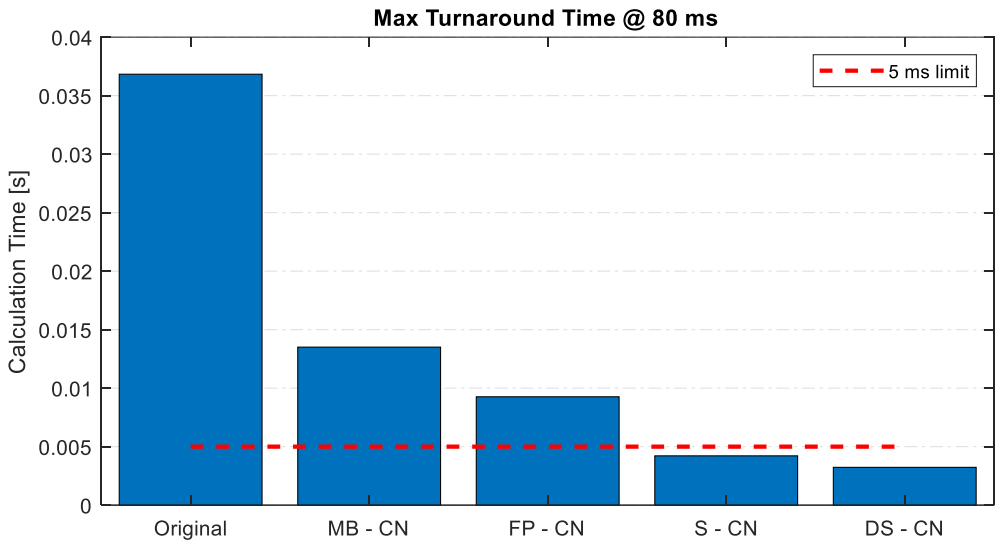

Figure 4.6: Bar graph of the maximum calculation time for each algorithm running with a model time step of Δt = 80 ms.

The tests are then repeated with the model step of 40 ms to see if there is any relative difference in calculation time with the results presented as Figure 4.7 – Figure 4.9. Overall the results are largely similar to the results from 80 milliseconds with the exception of FP – CN during the load drop from 250 A to 200 A. Here as evident in Figure 4.8, FP – CN takes longer than the modified bisection method (MB – CN) to calculate during the sharp transient event. Even though the technique once again quickly decreases in calculation time after resolving the immediate shock, it is largely overshadowed once again by S – CN and DS – CN as these techniques featured much smaller jumps in calculation time and appeared overall to perform more consistently. Figure 4.9 shows the maxima in calculation time for each method and as stated prior, only S – CN and DS – CN were capable of sub 5 millisecond calculation time.

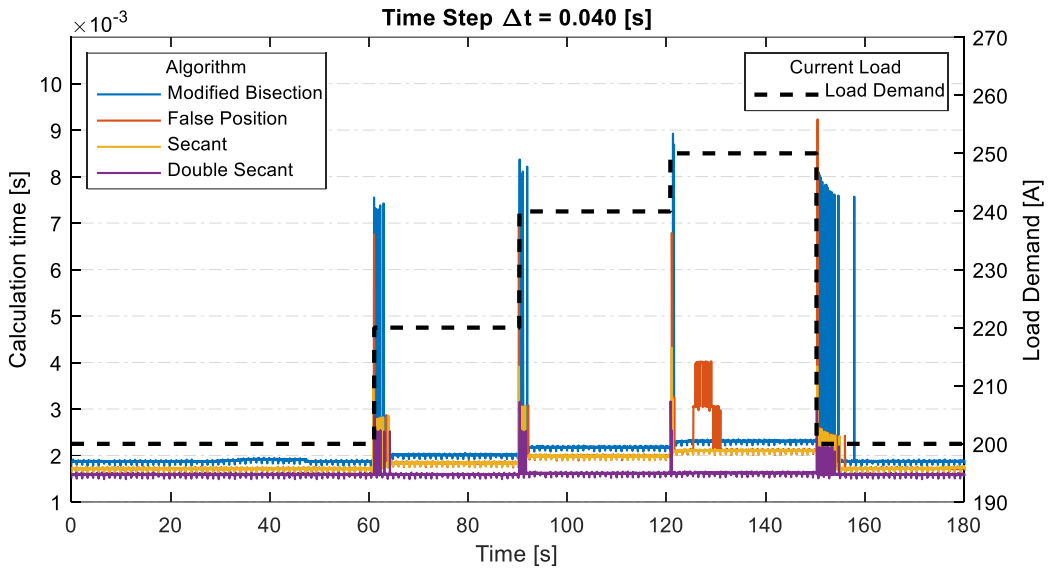

Figure 4.7: Plot of calculation time in seconds for different rootfinding methods during drastic load changes. Using Crank-Nicolson time marching scheme. Simulation running at Δt = 40 ms.

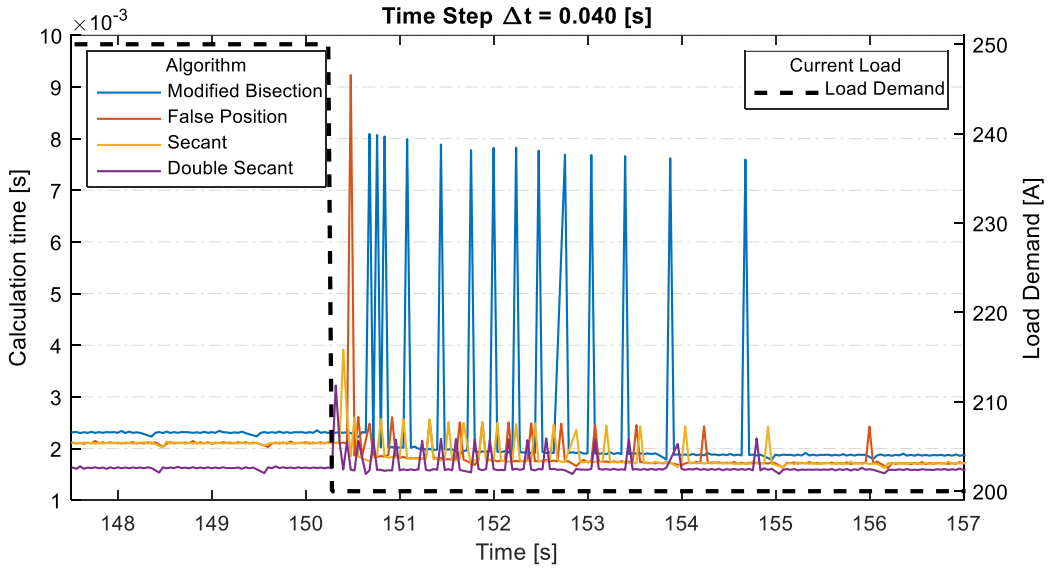

Figure 4.8: Zoomed in plot of calculation time in seconds for different rootfinding methods during drastic load changes. Using Crank-Nicolson time marching scheme. Simulation running at Δt = 40 ms.

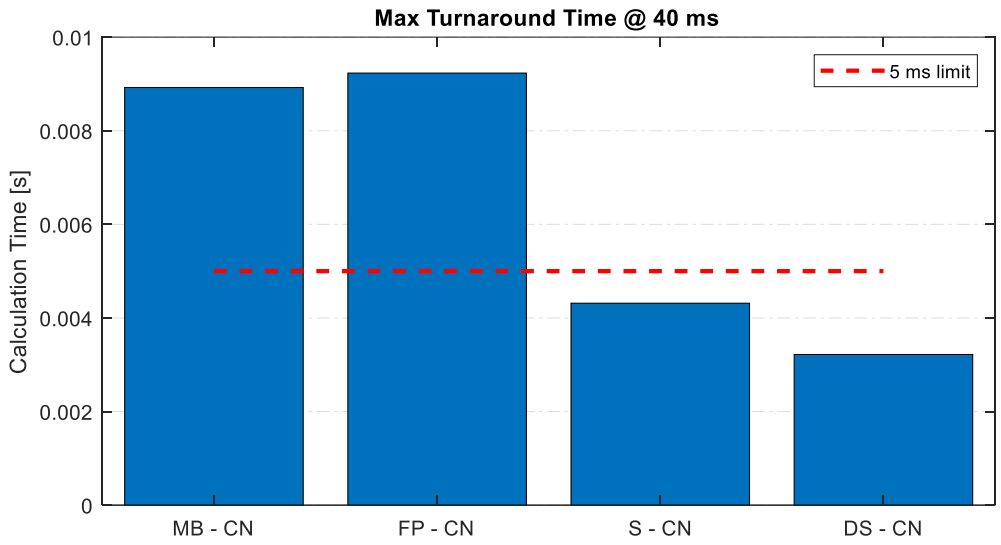

Figure 4.9: Bar graph of the maximum calculation time for each algorithm running with a model time step of Δt = 40 ms.

With successful tests at both 80 ms and 40 ms, the code was then set to the lowest time that could be set on dSPACE of 10 ms with the results shown below as Figure 4.10 – Figure 4.12. Despite the results indicating that sub 5 millisecond calculation time was possible, an overrun during the initialization phase led to the code failing to compile at a Δt of 5 ms. Later on, as will be discussed in Section 4.2.4, it was discovered that a set number of overruns could be set, allowing the code to compile despite overruns in the initialization and compilation of the code. Regardless, at this very small time step, the MB – CN algorithm fails to compile and can no longer run effectively, as it overruns too consistently to be safe to use in a real fully-coupled test indicated by some of the previous results, and as such is removed from testing. Likewise, FP – CN is near its performance limit as clearly visible in Figure 4.12, nearly reaching the threshold of 10 milliseconds of calculation time, despite still being capable of running at this small a time step.

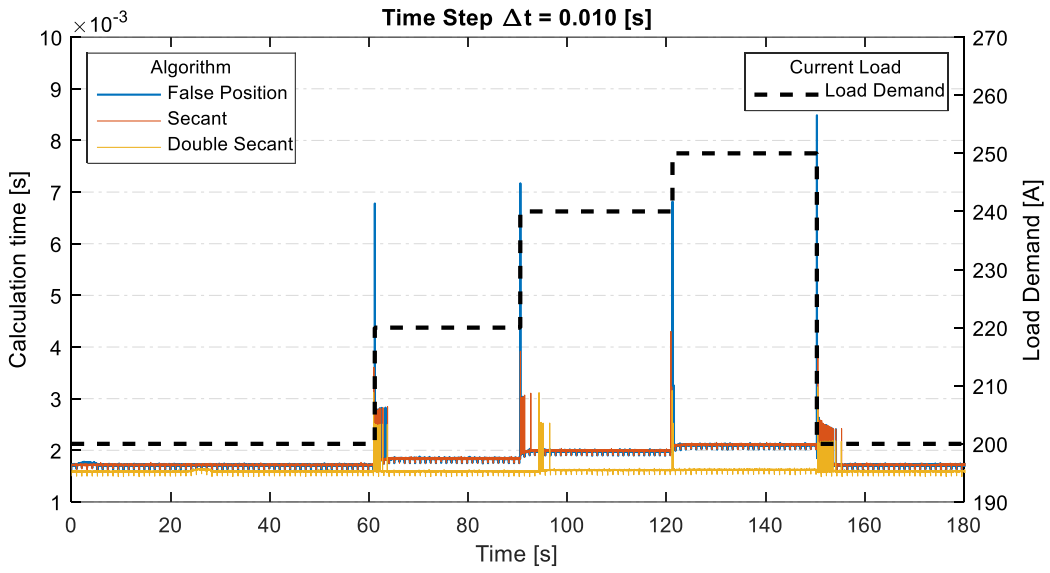

Figure 4.10: Plot of calculation time in seconds for different rootfinding methods during drastic load changes. Using Crank-Nicolson time marching scheme. Simulation running at Δt = 10 ms.

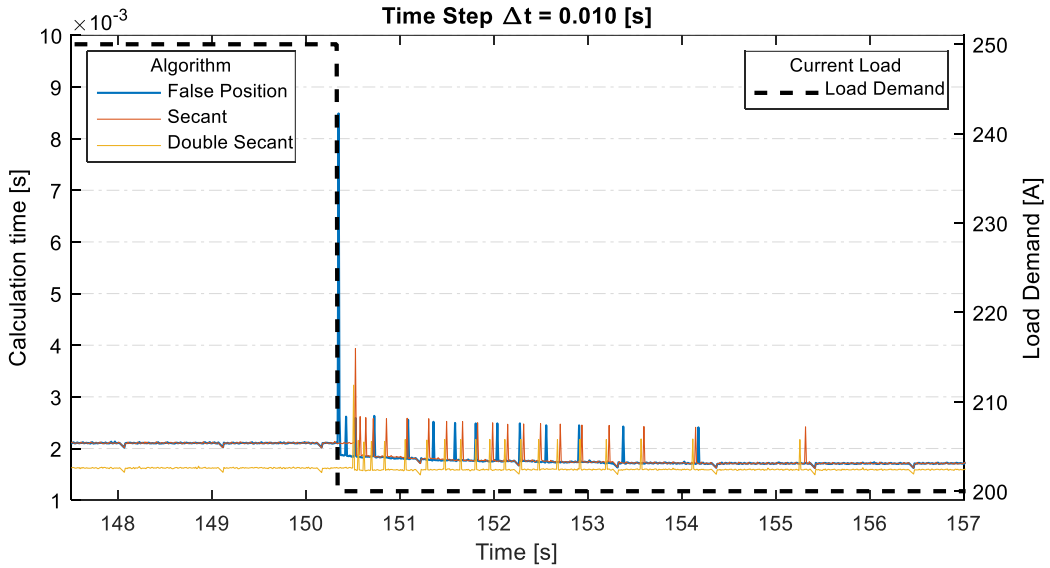

Figure 4.11: Zoomed in plot of calculation time in seconds for different rootfinding methods during drastic load changes. Using Crank-Nicolson time marching scheme. Simulation running at Δt = 10 ms.

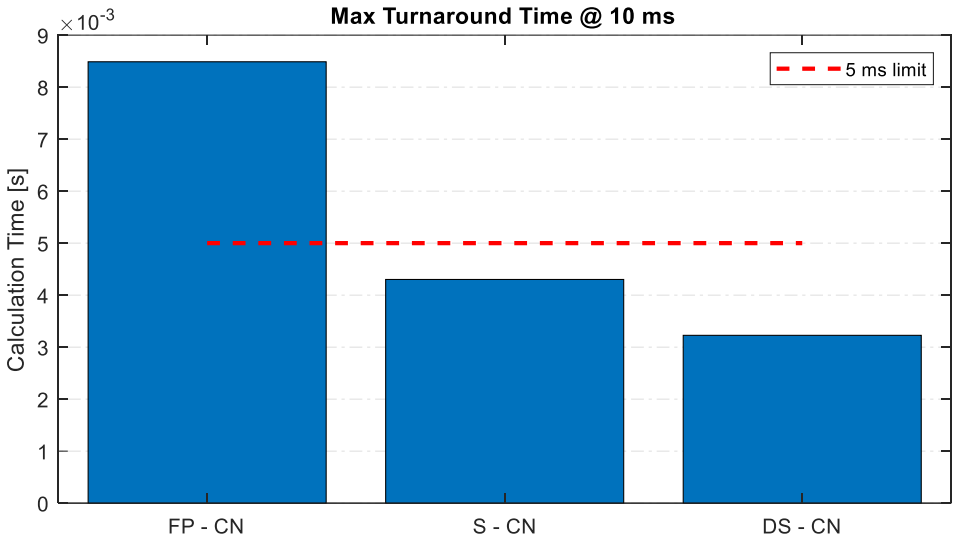

Figure 4.12: Bar graph of the maximum calculation time for each algorithm running with a model time step of Δt = 10 ms.

With the conclusion of all tests on the Crank-Nicolson based algorithms, all are plotted together in Figure 4.13 for the sake of comparison. Overall the results indicate that at any set Δt, the secant and double secant algorithms are capable of sub 5 millisecond calculation despite dSPACE indicating a failure to compile. As stated previously, later in development it was discovered that dSPACE's Control Desk software can be modified to allow for a preset number of overruns, aiding in mitigating the initialization errors where calculation time takes too long during the first run of the algorithm. Unfortunately, by this point the original ds1006 was already in the process of being replaced and could not be verified.

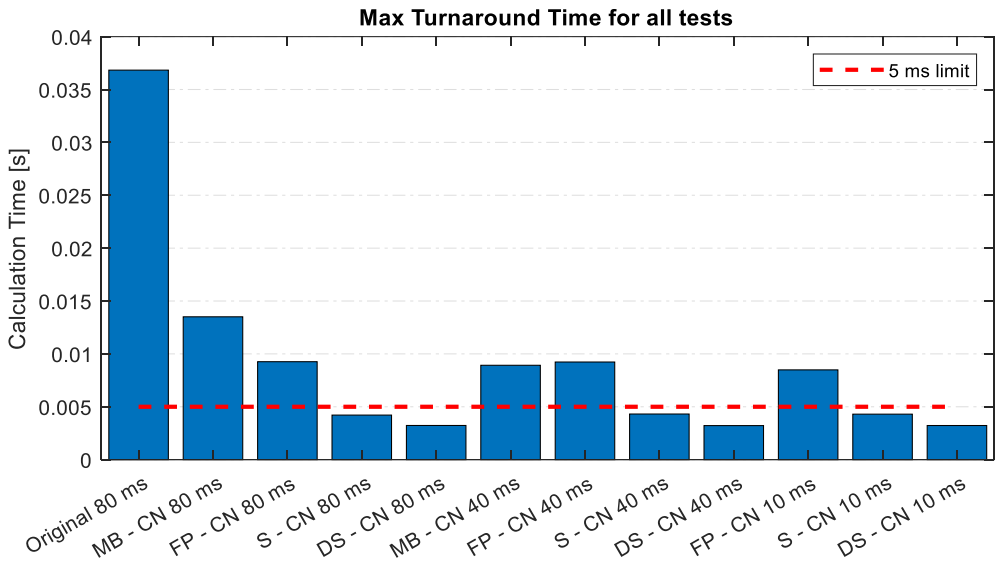

Figure 4.13: Compilation of bar graphs of the maximum calculation time for all algorithms using Crank-Nicolson (CN) for all different Δt.

With the conclusion of the Crank-Nicolson testing, next the Three-Point thermal algorithms were tested in the same manner as before. For these cases, only the extrema cases were investigated in order to quickly determine the general characteristics of the Three-Point thermal algorithm. Overall the results feature a similar pattern as the Crank-Nicolson set of results, shown below as Figure 4.14. For a full comparison, the Crank-Nicolson and Three-Point algorithm results are plotted together within Figure 4.15 in order to determine any differences between the two. For the most part the algorithms appear to perform comparably, with the exception of the "Original – TP" algorithm, which uses the original electrochemical algorithm and the new three-point time marching thermal scheme, at 80 milliseconds which performs slightly better than the original SOFC algorithm circa five percent from 36.8 milliseconds to 34.9 milliseconds. Once again, overall the time results seem more dependent on the electrochemical algorithm being used, with S – CN, DS – CN, S – TP, and DS – TP all being below the 5 millisecond calculation time threshold.

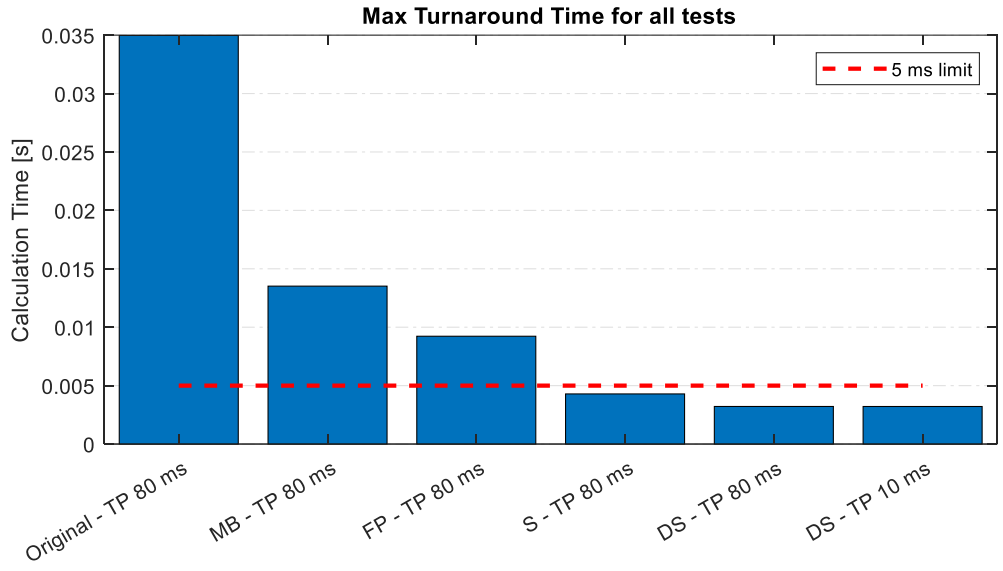

Figure 4.14: Compilation of bar graphs of the maximum calculation time for all algorithms using Three-Point (TP) for all different Δt.

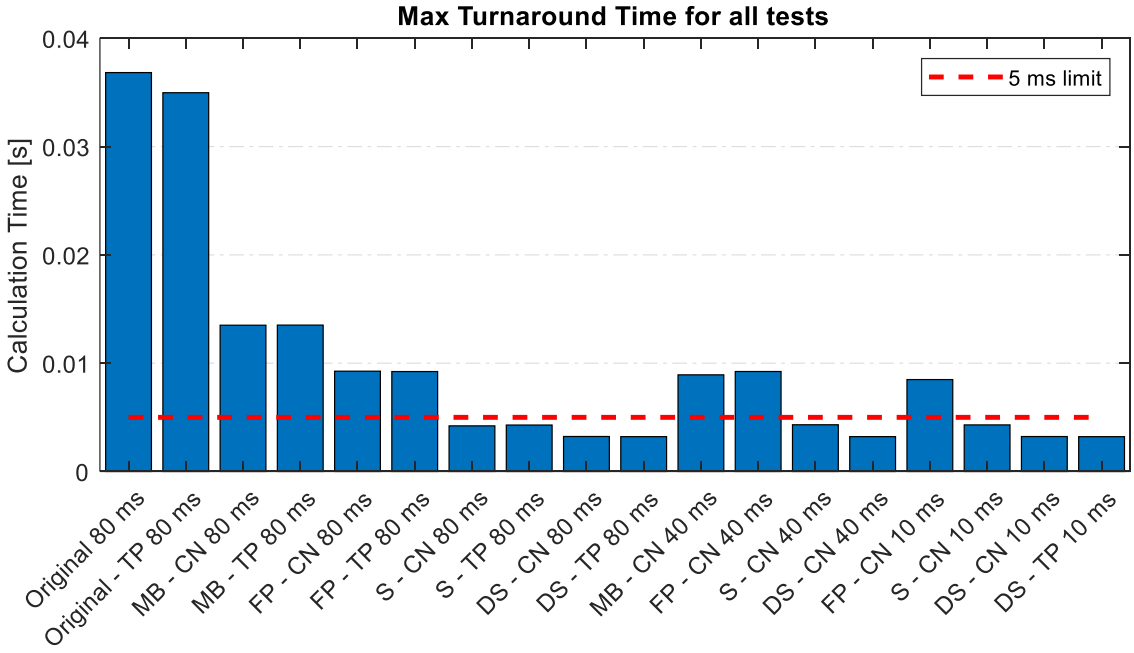

Figure 4.15: Compilation of bar graphs of the maximum calculation time for all algorithms.

4.2.2 Performance with Live Transient Inputs

A preliminary investigation into the performance of the SOFC code using live inputs from the Hyper turbomachinery was performed with a limited set of algorithms, to scope out the effect of noise being introduced into the new electrochemical algorithms with the results presented below as Figure 4.16 – Figure 4.18. For these tests, only the Crank-Nicolson thermal algorithm was used, as it was assumed that Three-Point would behave similarly, and likewise only a Δt of 10 ms was used, since it was apparent that this time could easily be met from the previous results. Overall the results were comparable once again to the previous results with the caveat that the resulting calculation times likewise appeared noisier. What is meant by this is that there were significantly more spikes in the calculation time plot, indicating that the code had to reconverge to its solution much more often, but seemed to stay in the same performance bounds overall with FP – CN nearing the 10 millisecond calculation time threshold, and S – CN and DS – CN comfortably remaining below a 5 millisecond calculation time.

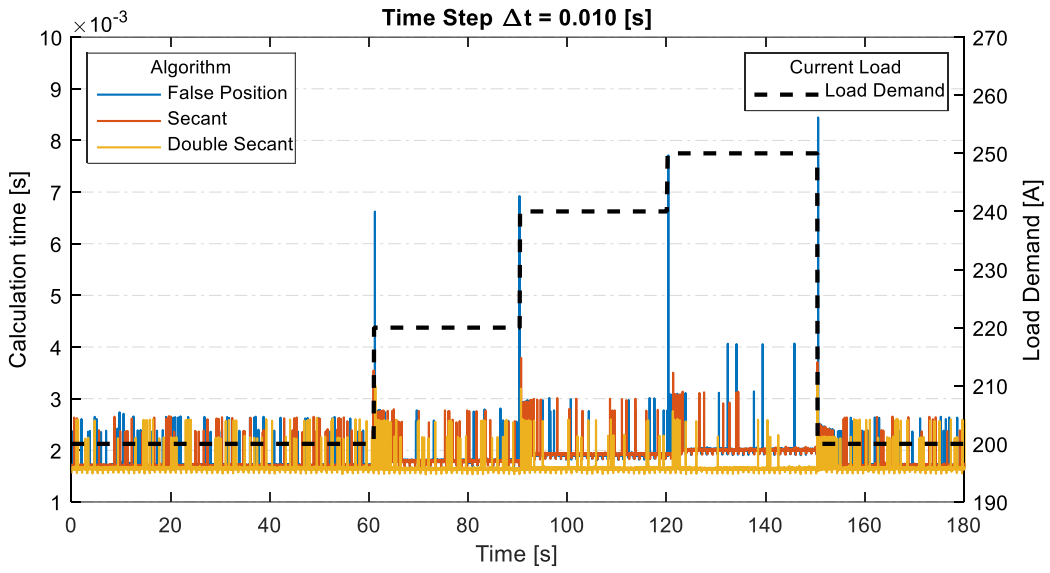

Figure 4.16: Plot of calculation time in seconds for different rootfinding methods during drastic load changes. Simulation running at Δt = 10 ms. Features live input data from turbomachinery.

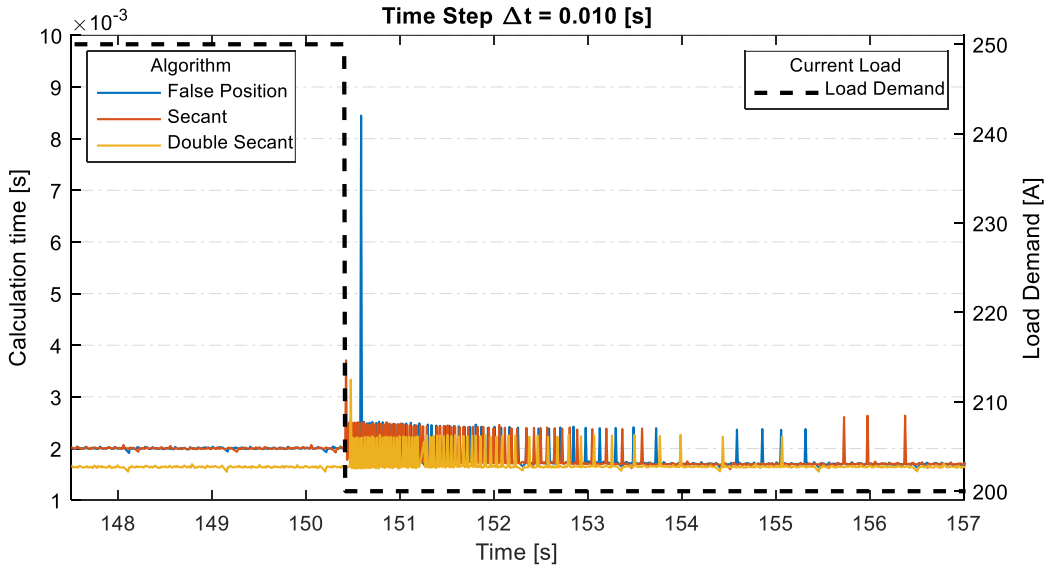

Figure 4.17: Zoomed in plot of calculation time in seconds for different rootfinding methods during drastic load changes. Simulation running at Δt = 10 ms. Features live input data from turbomachinery.

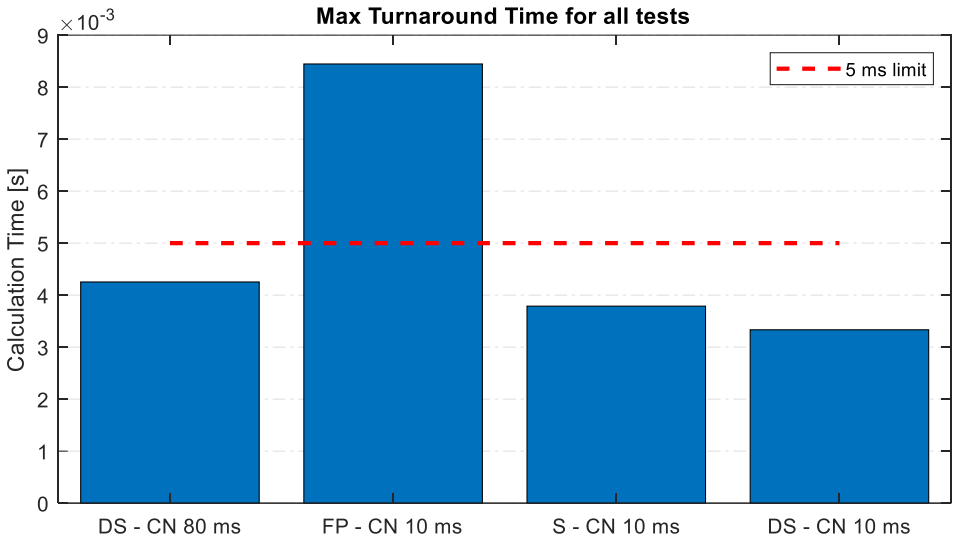

Figure 4.18: Compilation of bar graphs of the maximum calculation time for all algorithms using Crank-Nicolson (CN) for all different Δt. Features live input data from turbomachinery.

4.2.3 Fully Coupled Performance Tests

For the fully coupled Hyper tests the three main input parameters were changed and the aggregate results were used to determine the maximum turnaround time, i.e. the calculation time of the SOFC algorithm and any and all secondary calculations, as this was the limiting factor when setting the overall algorithm sample time. The actual test that was performed is defined below using Table 4.1 and features changes to the three main inputs to the SOFC code.

Table 4.1: Program for fully coupled testing

Stimuli	Before	After	Duration
1. Fuel Composition	Syngas	17% Methane 83% Steam	60 sec
2. Hot Air Bypass	25% Open	80% Open	60 sec
3. Load Change	200 A	100 A	60 sec
4. Fuel Composition	Syngas	13.7% Methane 86.3% Steam	30 sec
5. Hot Air Bypass	25% Open	40% Open	30 sec
6. Load Change	200 A	210 A	30 sec

The first set of stimuli (1-3) were performed with only live inputs as they were exaggerated analogues of the proceeding second set of stimuli (4-6) that were subsequently performed while fully coupled. This was performed to ensure a relative degree of operational safety so as to not go outside the operating envelop of the turbomachinery which could have resulted in either surge or stall and subsequently risk heavy damage to the Hyper machinery. Likewise, after the prescribed duration of each stimulus, the inputs to Hyper were set back to the starting operating values. The tests were performed with both the Crank-Nicolson and Three-Point thermal algorithms first at 20 milliseconds, then repeated again at 10 milliseconds and represented by Figure 4.19 and Figure 4.20 respectively. Overall the results indicate a strange contradiction between the Crank-Nicolson results and Three-Point results that will be discussed further in the succeeding paragraph.

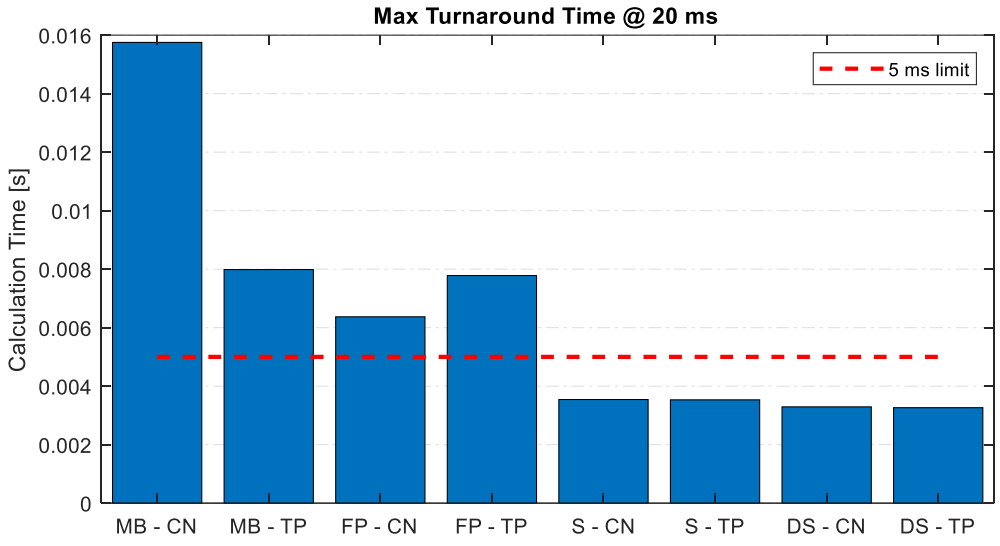

Figure 4.19: Bar graph of the maximum calculation time for each algorithm running with a model time step of Δt = 20 ms. Features full coupling to turbomachinery.

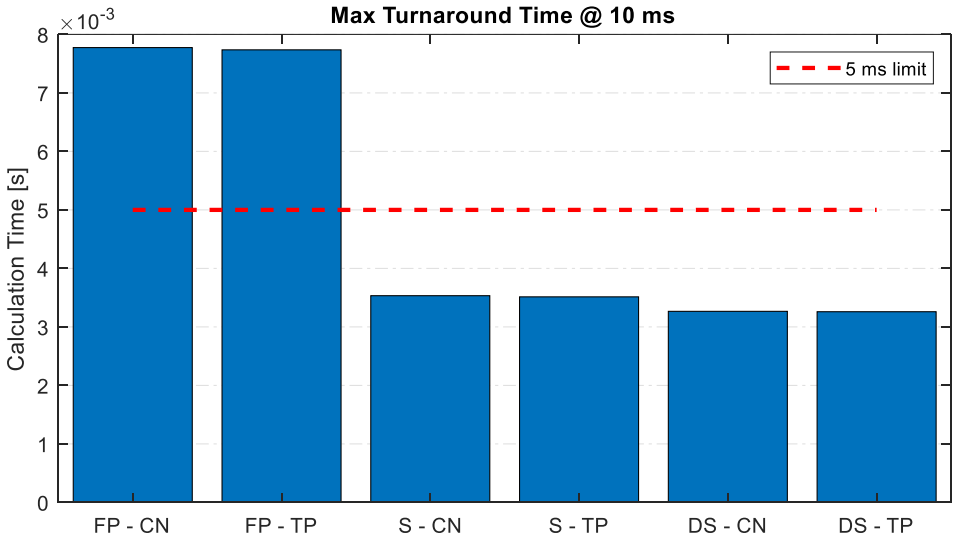

Figure 4.20: Bar graph of the maximum calculation time for each algorithm running with a model time step of Δt = 10 ms. Features full coupling to turbomachinery.

For full comparison, the results are all compiled together as Figure 4.21 to illustrate the differences between the Crank-Nicolson and Three-Point results. An interesting

contradiction is apparent, where CN and TP both appear to perform better than the other method depending on the electrochemical algorithm being used. Most dramatic is the difference in results between MB – CN and MB – TP at Δt of 20 milliseconds. Here MB – TP appears to take half the time of MB – CN to calculate which would be strong evidence in support of using the TP thermal algorithm. Listed as a percentage the difference is 49.3% at 15.75 milliseconds versus 7.985 milliseconds. However, when using the false position electrochemical method, FP – CN appears to calculate significantly faster than FP – TP when using a Δt of 20 milliseconds, directly contradicting the first set of results, with FP – TP taking 22.1% longer to calculate than FP – CN. What can only be inferred then, is that the calculation time is severely sensitive to external stimuli and the synergy between the two main algorithms. Despite these particularities however, overall the trend remains with the secant and double secant methods meeting the goal of sub 5 millisecond calculation time regardless of thermal algorithm or Δt.

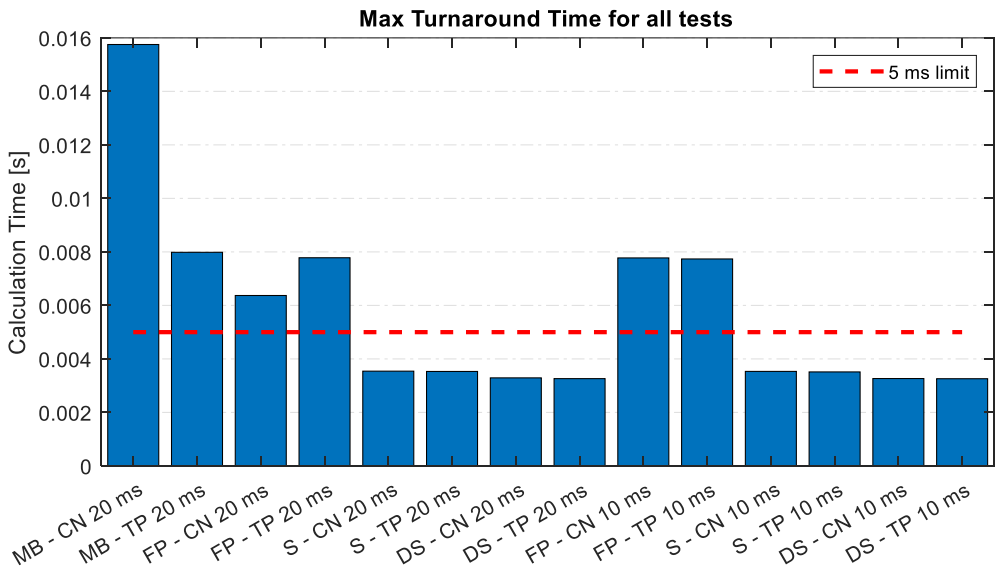

Figure 4.21: Compilation of bar graphs of the maximum calculation time for all algorithms. Features full coupling to turbomachinery.

To conclude the principal investigation, verification of the numerical methods was performed using the fully coupled data sets to ensure accuracy despite such radical changes to the numerical platform. These involved taking the results of the original SOFC algorithm, which used the original electrochemical algorithm along with the original tight tolerances and the Crank-Nicolson time marching scheme, and comparing them to the results of the new higher order electrochemical algorithms combined with the Three-Point thermal algorithm. The primary parameters investigated were the solid temperature profile and the current density distribution at steady state, as well as the total heat generation of the SOFC during a sharp transient event.

The steady state results are presented first with Figure 4.22 displaying the solid temperature profiles while Figure 4.23 plots the relative error between the solid temperature profiles. Likewise, Figure 4.24 displays current density profiles and with Figure 4.25 showing the relative error of those current density profiles. For brevity, the results of the original method (Original - CN) are compared to all the electrochemical algorithms but using only the Three-Point thermal algorithm. The thought was that deviations stemming from either the new electrochemical formulations or the thermal algorithm would distinctly be visible from either the temperature results or the current density profiles. Overall the general trend is that the new algorithms resolve to the same steady state solutions, with the temperature profiles showing a maximum relative error between the new methods and the original code of only around 2.5e-4 or 0.025%. Likewise, the current density profiles also only see a relative difference as high as 1.6e-3 or 0.16%.

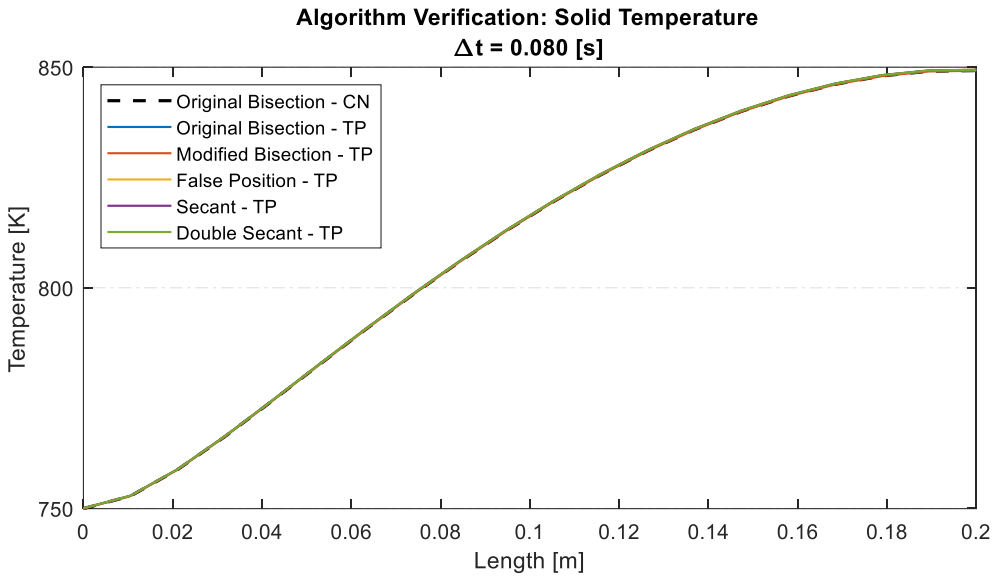

Figure 4.22: Temperature profiles for solid region of SOFC. Illustrate that differences to electrochemical or thermal algorithm do not affect the results when resolving the temperature profiles.

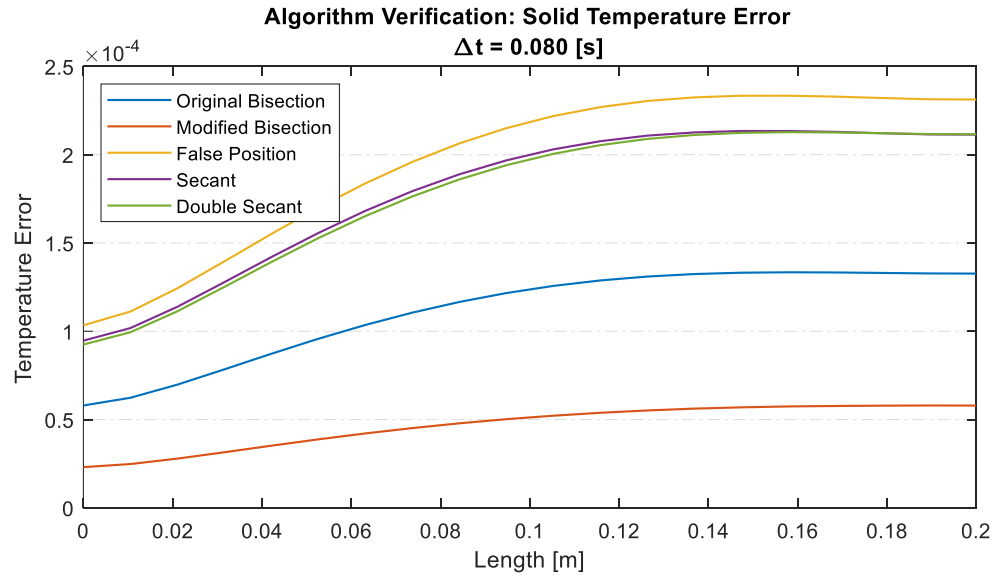

Figure 4.23: Plot of relative error of temperature profiles as compared to the original SOFC code provided by NETL. Curves illustrate a difference of less than 1% for the different algorithms at steady state.

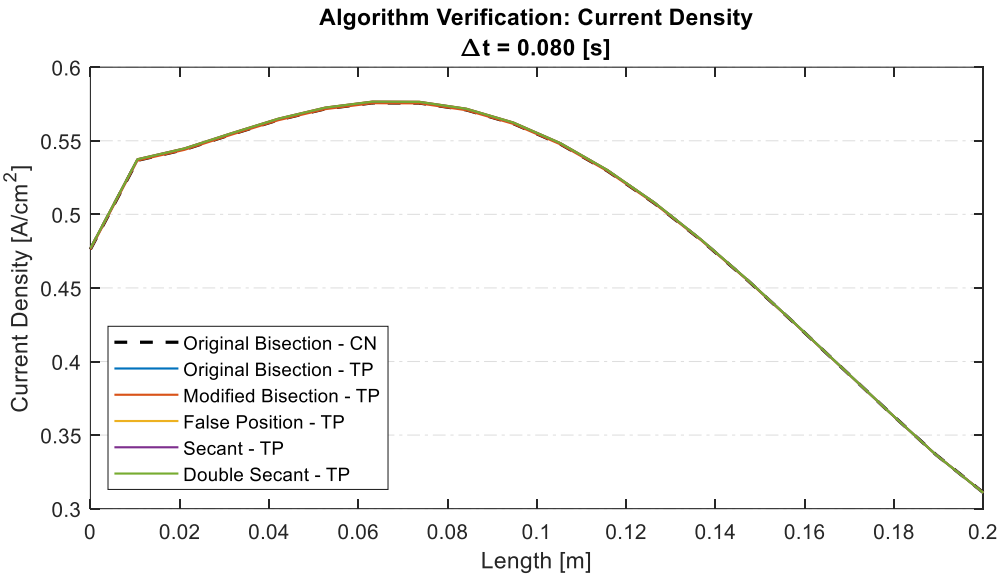

Figure 4.24: Current density profiles for SOFC. Illustrate that differences to electrochemical or thermal algorithm do not affect the results when resolving the current density profiles.

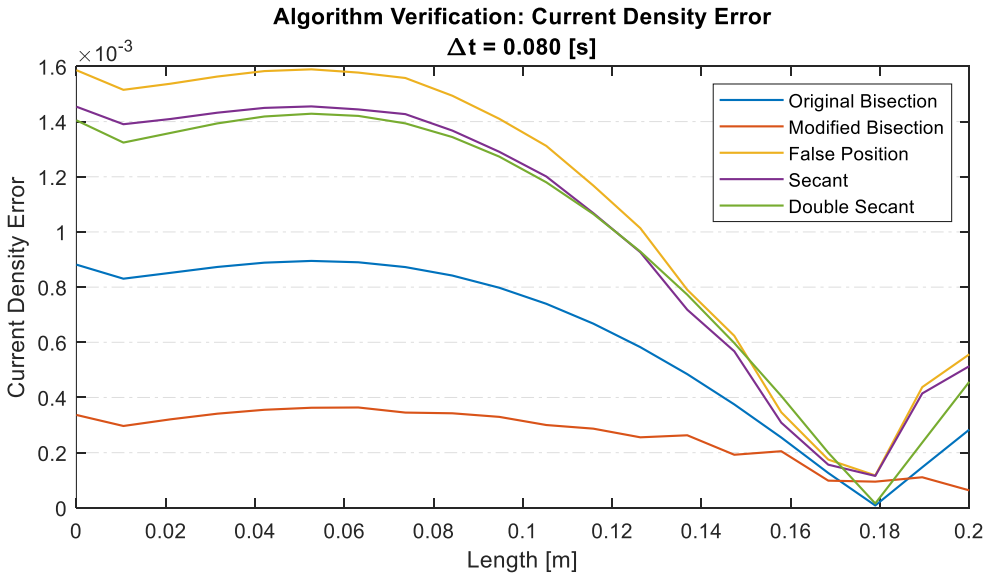

Figure 4.25: Plot of relative error of current density profiles as compared to the original SOFC code provided by NETL. Curves illustrate a difference of less than 1% for the different algorithms at steady state.

Lastly, the total heat generation of the SOFC is plotted for the different algorithms during the transient event defined by stimulus 1 in Table 4.1, a change from Syngas to

Humidified Methane. The results are presented as Figure 4.26 with the relative error normalized to the original model presented as Figure 4.27. From the evolution of the heat generation for the different algorithms, it can be visibly seen that the profiles follow the same general trend, with an initial sharp decrease in heat output followed by a smaller recovery in heat generation over a longer period of time. During this recovery period, a slight deviation can be seen between the data from the original code (Original - CN) and the new models. While somewhat worrisome, the plot of the relative error as compared to the original indicates that this deviation is still below 1%. Additionally, a key distinction between the original results and the three-point based data sets is that the original test was performed on a different test date. Given that the turbomachinery of the Hyper facility is fueled using natural gas, due to natural day to day changes in fuel composition, slight differences in baseline and operating conditions can result. This change in fuel composition could help explain the small difference in the transient response of the original code and the new models, where the new models all align very closely to one another, but all are slightly different compared to the original model results.

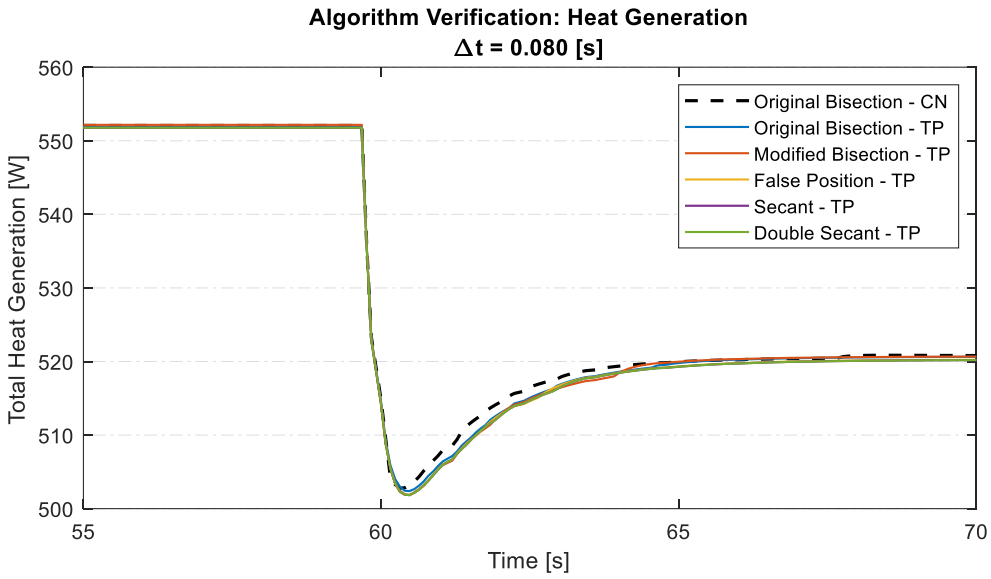

Figure 4.26: Total heat generation from SOFC during change in fuel composition from Syngas to Humidified Methane. Slight deviation is seen in region after initial shock, however the same general trend is present between all algorithms.

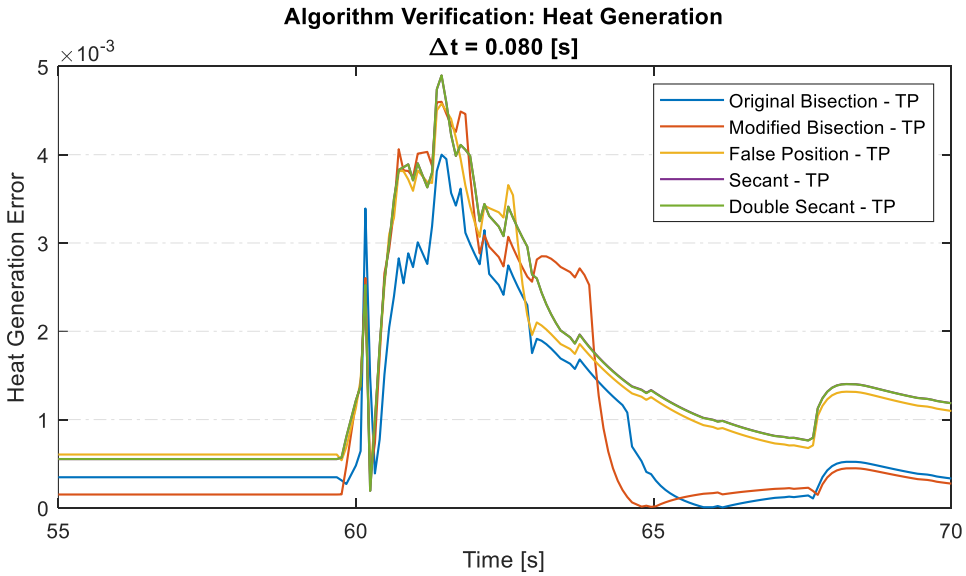

Figure 4.27: Relative error in total heat generation as compared to original SOFC code from NETL. Overall results indicate relative error increases during heavy transience but overall still remains below 1% during entire event.

4.2.4 Performance on Upgraded dSPACE Platform

Towards the end of the onsite investigation at NETL, several hardware issues led the team to replace the entire dSPACE platform with a newer processing system, the dSPACE SCALEXIO specifically powered by a DS6001 Processor Board. The same tests that were performed in Section 4.2.3 were repeated for the new dSPACE platform first at a Δt of 20 milliseconds so as to be comparable to the original dSPACE results for fully coupled operation, then at the project goal of $\Delta t = 5$ ms. The results are presented below as Figure 4.28 and Figure 4.29 for $\Delta t = 20$ ms and 5 ms, respectively. The 20 millisecond results were fairly limited in scope, as the algorithms were fully expected to function on the new dSPACE platform and were used to be able to quickly characterize the performance range of the algorithms and compare such to the old platform. A significant reduction in calculation time was still seen between the modified bisection methods and the double secant methods. An interesting development was seen for the 5 millisecond Δt results, where the modified bisection results took longer than the 20 millisecond results overrunning consistently, with a max turnaround time of approximately 6.5 milliseconds. This did however illustrate the capability of dSPACE to resolve overruns, as long as they don't happen more than the specified number of allowable overruns. It seemed that the higher-order rootfinding schemes were absolutely necessary for ensuring a sub 5 millisecond calculation time for the entire simulation.

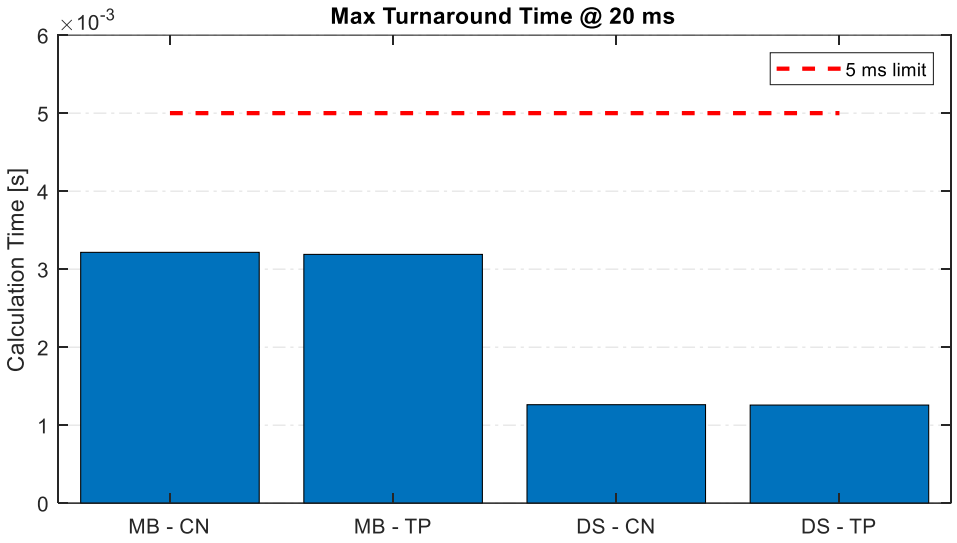

Figure 4.28: Bar graph of the maximum calculation time for each algorithm running with a model time step of Δt = 20 ms. Features full coupling to turbomachinery on new dSPACE platform.

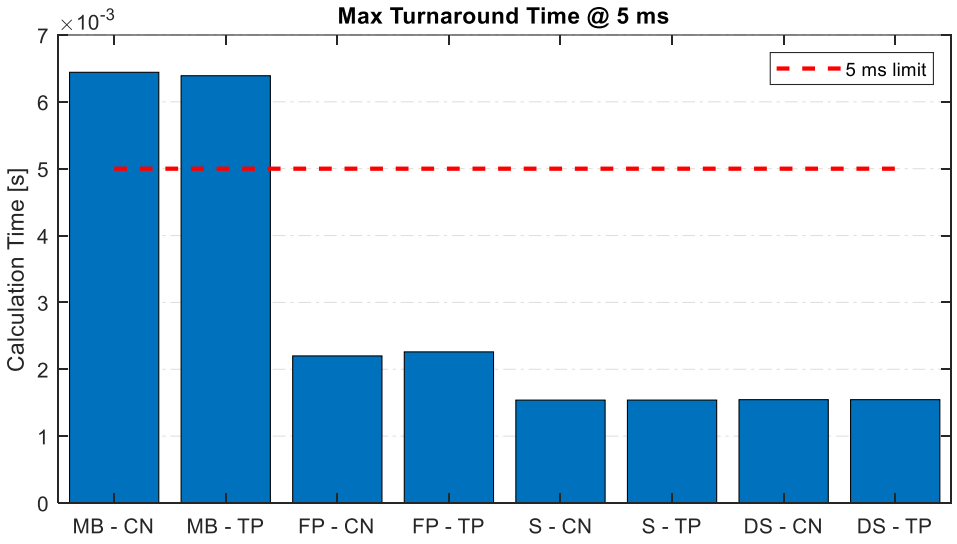

Figure 4.29: Bar graph of the maximum calculation time for each algorithm running with a model time step of Δt = 5 ms. Features full coupling to turbomachinery on new dSPACE platform.

The results on the new dSPACE platform were all compiled as shown in Figure 4.30 to illustrate the differences between the Crank-Nicolson and Three-Point results. Overall, the previous trends between Crank-Nicolson and the Three-Point scheme don't appear to materialize in a significant fashion with nearly identical results between the thermal algorithms. This is not necessarily a bad result, as this allows the Three-Point method, which is fully implicit and unconditionally stable along with being second order accurate in time [28], to be used without fear of significantly increased computational time. As stated in the preceding section however, the modified bisection methods at a 5 millisecond time step appear to fail at maintaining a sub 5 millisecond calculation time. Likewise, in general the 5 millisecond time step results took longer, with the MB methods taking almost twice as long, and the DS methods take only slightly longer at around 22.1%.

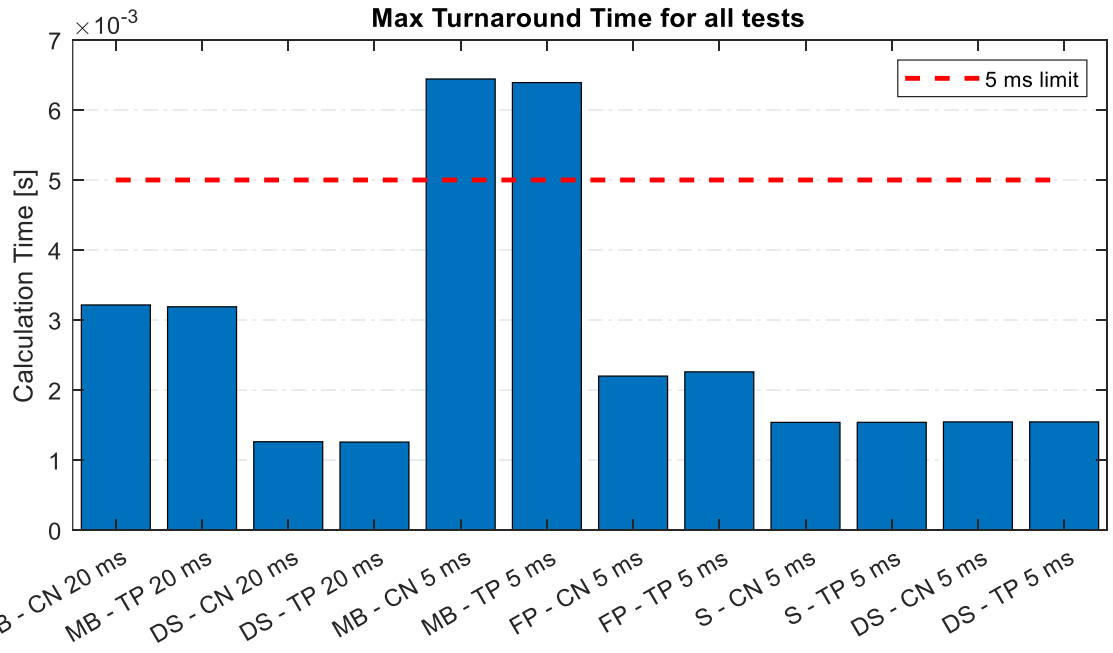

Figure 4.30: Compilation of bar graphs of the maximum calculation time for all algorithms. Features full coupling to turbomachinery on new dSPACE platform.

To fully conclude the study, the results from the new dSPACE platform were then compared against the old dSPACE platform and presented below as Figure 4.31. As was to be expected, a significant reduction in calculation time is present for all cases ranging from as low a reduction as 60.1% to as high as 79.5% with a calculation time as low as 1.25 milliseconds for the DS – CN and DS – TP methods.

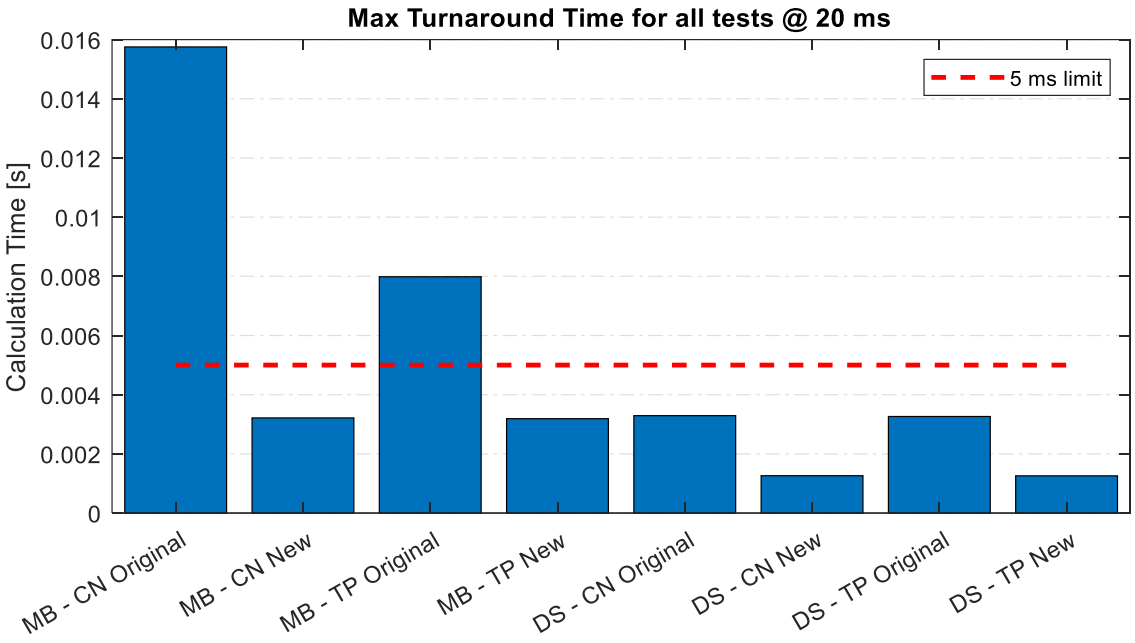

Figure 4.31: Compilation of bar graphs of the maximum calculation time for algorithms running on original dSPACE hardware versus new dSPACE running with a model time step of Δt = 20 ms. Features full coupling to turbomachinery on new dSPACE platform.

CHAPTER 5. PRELIMINARY AND FUTURE WORK

5.1 Sensitivity Analysis

Late in development, it was discovered that certain input fuel streams (the particular case being 25% Methane, 75% steam) result in a large energy imbalance at steady state as defined by Equation (5.1) with the relative error being defined as Equation (5.2). The equations used to determine the energy imbalance are (5.3) for the total heat generated by the cell, and (5.4) for the sensible heating done to the gaseous stream exiting the fuel cell.

$$\dot{Q}_{Imbal} = \dot{Q}_{Gen} - \dot{Q}_{Sens} \tag{5.1}$$

$$\epsilon_{Imbal} = \frac{\left|\dot{Q}_{Gen} - \dot{Q}_{Sens}\right|}{\left|\dot{Q}_{Gen}\right|} \tag{5.2}$$

$$\dot{Q}_{Gen} = \sum_{i=1}^{20} \dot{Q}_{Gen}(i) \tag{5.3}$$

$$\dot{Q}_{Sens} = \dot{m} c_{p,g} (T_{g,outlet} - T_{g,inlet}) \tag{5.4}$$

Due to the high concentration of methane in the intake gas stream, it is known that reformation will occur in the initial portion of the cell, and that this reformation reaction is highly endothermic leading to a very complex temperature profile in the gas stream seen below as Figure 5.1.

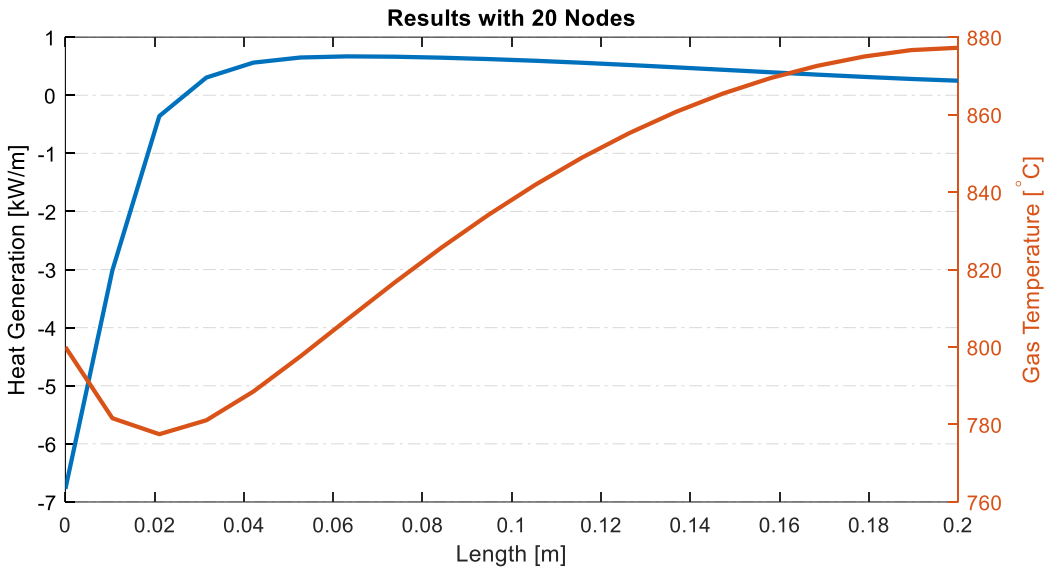

Figure 5.1: Plot of Heat Generation profile throughout the cell along with the Gas Temperature profile. Clear evidence of strong endothermic reaction in the first quarter of the fuel cell resulting from the reformation of methane. This results in the cooling of the gas stream in the first quadrant of the cell.

5.1.1 Initial Results

A simple initial investigation was performed in MATLAB to determine the sensitivity of the algorithm to the mesh resolution in the extreme case of excessive fuel reformation occurring in the fuel cell. The results of the preliminary investigation are below as Figure 5.2 – Figure 5.4. Overall the results indicate that the gas stream is highly sensitive to the model resolution as evident in Figure 5.2, with the temperature profile experiencing significant deviation until a resolution of at least 250 nodes. Likewise Figure 5.3, the plot of the two relevant heat terms, seems to reinforce the hypothesis that the gas temperature requires a higher resolution to properly reproduce the correct steady state profile as the sensible heat values diverge more rapidly than the heat generation at low resolutions. Additionally, as seen in Figure 5.4, the relative error of the energy balance decreases at a linear rate, which coincides with the first order discretization of the gas stream, and

likewise informs that a high discretization must be used to reduce the error to an acceptable value.

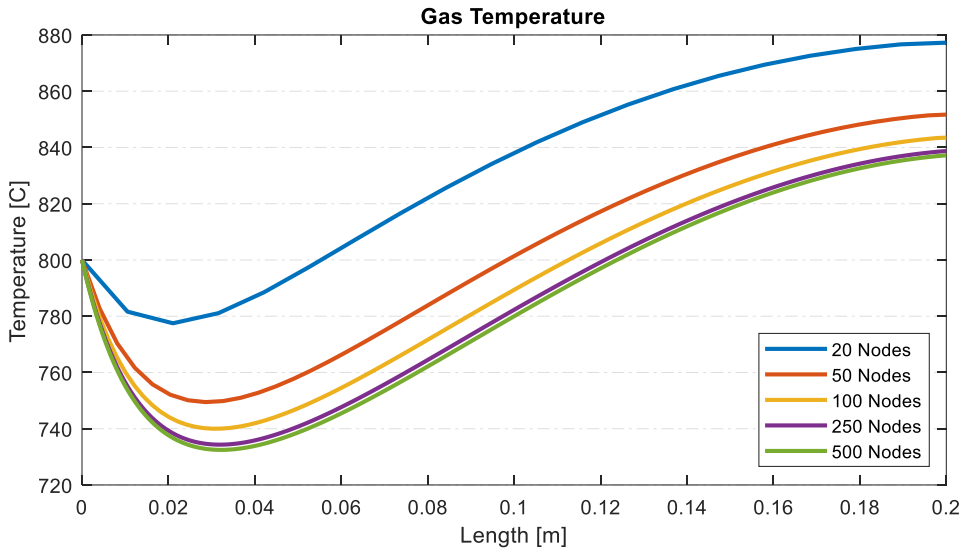

Figure 5.2: Plot of gas temperature profile for increasing resolution. Clear evidence of strong endothermic reaction in the first quarter of the fuel cell resulting from the reformation of methane. This results in the cooling of the gas stream in the first quadrant of the cell.

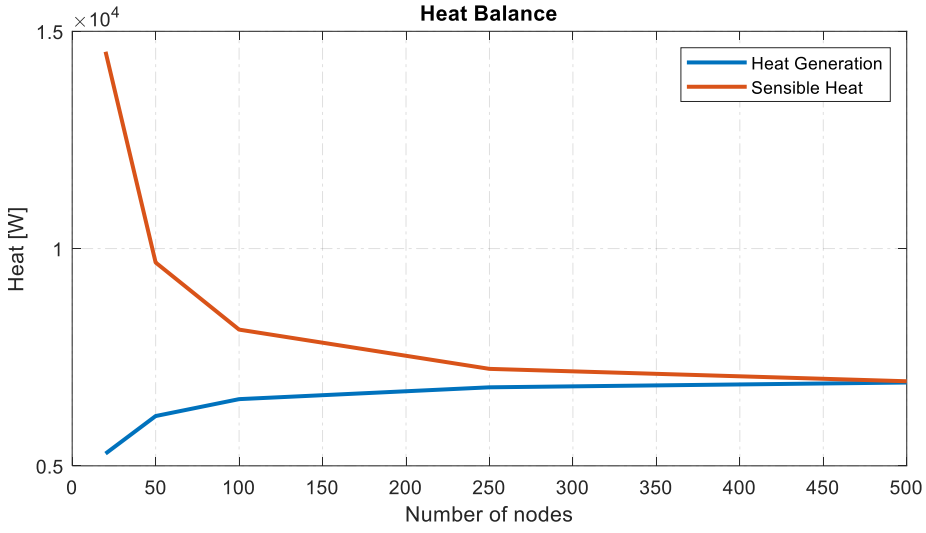

Figure 5.3: Plot of heat generation and sensible heat as model resolution increases. Evident that sensible heat is much more sensitive to resolution than heat generation in the cell.

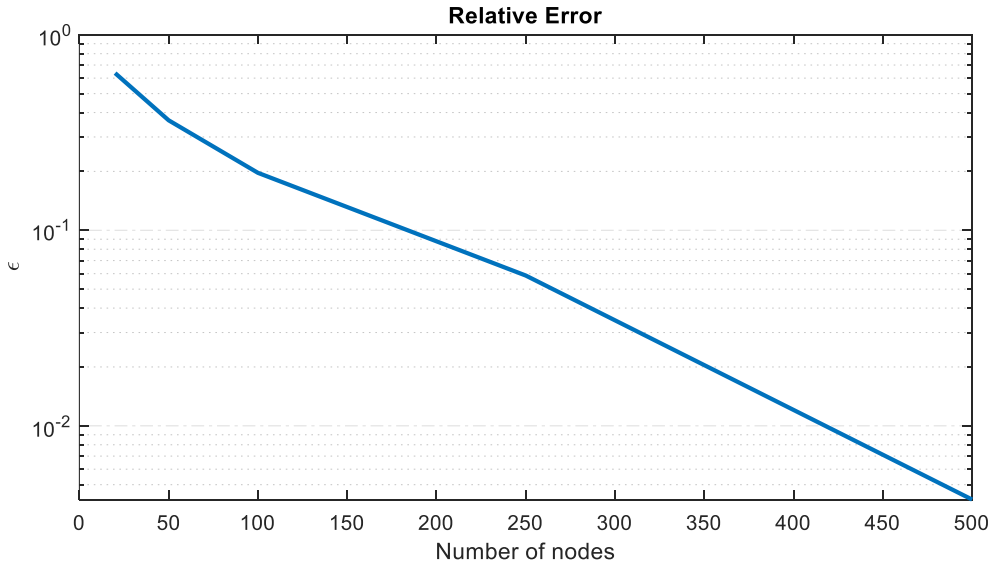

Figure 5.4: Plot of the relative error between the heat generation and the sensible heat. Overall a linear (1st order) decrease in heat imbalance can be deduced.

5.1.2 Impact of Resolution on Transient Timescales

An opportunity to test the new algorithms at a higher discretization became available, at which the characteristics of calculation time versus resolution were able to be investigated at least briefly. The results of the test are displayed individually in Figure 5.5, and are compared to the original 20 nodes results calculated on the original dSPACE system in Figure 5.6. Due to time constraints and likewise since the main difference in calculation time was expected to result from changing the electrochemical algorithm, the Three-Point thermal algorithm was omitted from investigation, testing only the Crank-Nicolson based configurations. The test performed for these higher resolution studies was the same offline test from Chapter 4.2.1. The results from Figure 5.5 show that increasing the resolution directly affects the calculation time of the model, and substantially so with

only the Double Secant (DS) method remaining at a sub 5 millisecond run time. Additionally, further increasing the resolution from 50 nodes to 100 shows more than a twofold increase in calculation time, showing that very high resolution cannot be achieved without allowing for a longer run time and subsequently a slower sampling rate. As compared to the original set of results however, the increases in the efficiency of the SOFC algorithm are still readily apparent if not more so, and especially for the high resolution cases. Even for the 100 node cases, the calculation times were still significantly lower than for the original model, meaning that for extreme cases, the resolution can be increased fivefold or more, and still operate at the original sampling rate of 80 milliseconds. In fact, for the Double Secant method, the 100 node case still operates between 5 to 10 milliseconds, meaning that a compromise between transient and spatial resolution can be done that still results in significantly better resolution for both.

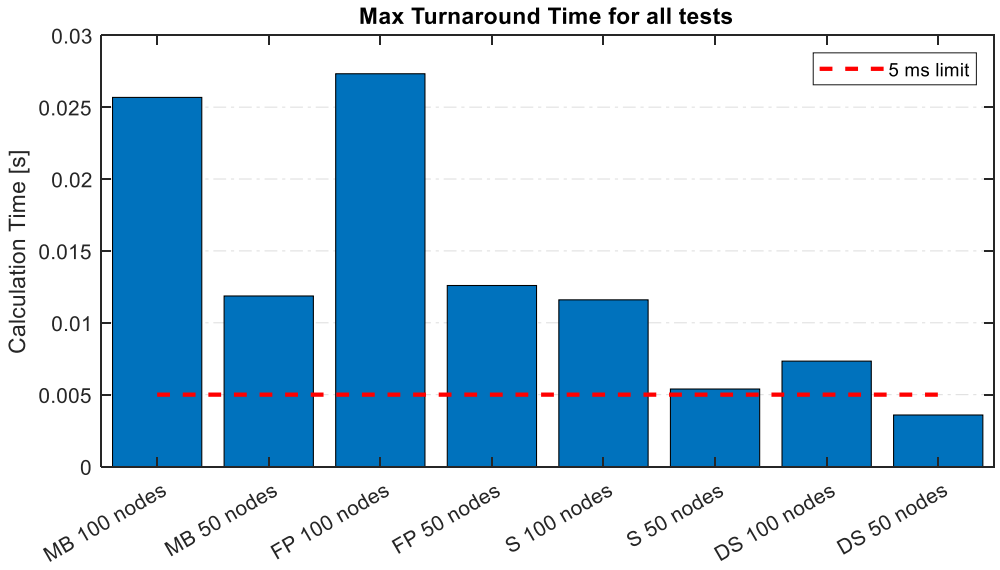

Figure 5.5: Calculation time of the different electrochemical algorithms at high resolution. Resolution increases directly affect calculation time, almost doubly so between 50 and 100 nodes. Only the DS method at 50 nodes can maintain sub 5 millisecond calculation time.

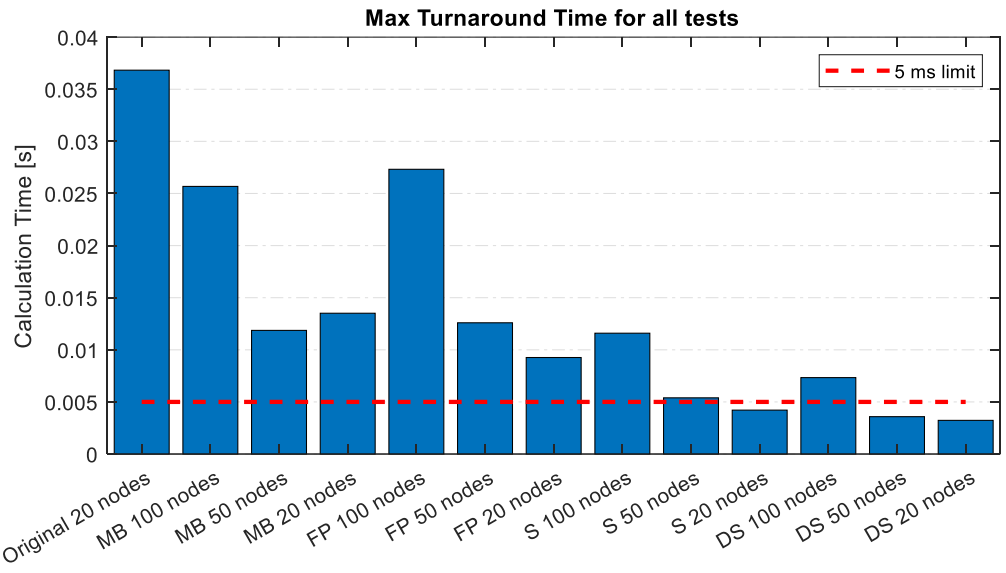

Figure 5.6: Calculation time at different resolutions as compared to the 20 node results. Resolution increases directly affect calculation time. Only the DS method at 50 nodes can maintain sub 5 millisecond calculation time.

At least in the short term, it appears as if increasing the resolution of the model, even if only to 100 nodes, helps mitigate the mesh sensitivity to some degree. Likewise, the new dSPACE platform can accommodate a significant increase in the resolution given that the appropriate time constraints are taken. Ultimately, however, a more robust solution will be necessary for resolving the complex behavior.

5.2 Updating Gas Stream Discretization and Adaptive Variable Discretization

Two possible outlets of investigation for resolving the cases from section 5.1 with severe generation terms and complex temperature profiles include discretizing the governing equation of the gas stream with a different derivation better suited for pure convection, and also adaptive variable discretization. Both lend themselves well to solving the governing equation for the gas stream given below as Equation (5.5).

$$\rho c_v A_c \frac{\partial T_g}{\partial t} + \rho c_p U A_c \frac{\partial T_g}{\partial x} = hP(T_s - T_g) \tag{5.5}$$

This governing equation is a hyperbolic partial differential equation, which corresponds to pure convection absent of any physical diffusion. Resolving this kind of behavior using a finite difference model results in complications when trying to accurately resolve the fluxes at each node face. Essentially, coarse meshes will result in high amounts of numerical dissipation as is expected from advection-dominated equations [29]. This negatively affects the energy balance as the advected quantity, which in this case is thermal energy, dissipates as it passes from one node to another resulting in unaccounted for energy.

To this end the first proposed solution is to replace the original governing equation or at least the method employed to determine the fluxes at each cell face. A sample form of the new governing equation could reincorporate the effects of thermal diffusion presented below as Equation (5.6)

$$\rho c_v A_c \frac{\partial T_g}{\partial t} + \rho c_p U A_c \frac{\partial T_g}{\partial x} = A_c \frac{\partial}{\partial x}\left(k \frac{\partial T_g}{\partial x}\right) + hP(T_s - T_g) \tag{5.6}$$

With this derivation, different approximations could be used to attempt to better simulate the effects of thermal transport throughout the domain, examples of which include the Hybrid Exponential scheme from Spalding, a Power-Law approximation, or the Lax-Wendroff scheme, etc [27]. Otherwise, as is done in computational fluid dynamics problems, higher order flux approximation schemes can be employed to maintain the

sharpness of the fluxes at the cell faces, instead of simple upwinding of the fluxes, with the aim to reduce dissipation of large gradients.

Secondly, a nonuniform discretization of the computational domain can also be implemented. The main idea behind this approach is that the highest temperature gradients, and subsequently the most complex behavior in the fuel cell, typically occur at the front half of the cell. Likewise, there are different methods that can be utilized from a simple subdivision of nodes done manually in the front of the fuel cell, to adaptive methods that increase resolution depending of the severity of the temperature gradient. Regardless of the exact approach, the requirement that calculation time remain as low as possible, means that targeted approaches such as this propose an alternative that can resolve the overall fuel cell with higher accuracy without the need to excessively increase the resolution of the model, focusing instead on the problematic areas of the domain.

5.3 Object Oriented Programming (OOP)

A particular interest of late is to convert the SOFC code into an object-oriented programming (OOP) architecture due to the need for increased modularity in the SOFC platform. This is particularly pertinent for rapid testing of substantially different fuel cell designs and alternative physical formulations, with examples ranging from using the Three-Point thermal algorithm versus Crank-Nicolson, to the new flux estimation methods and domain discretization methods proposed in the above section. An example layout of the modules is shown below as Figure 5.7.

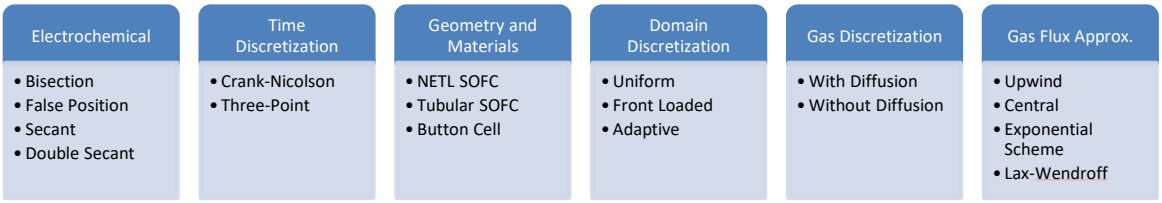

Figure 5.7: Illustration of the individual modules that would allow for quickly reconfiguring Hyper for different SOFC designs and likewise the implementation of different physical models.

Upon moving to this style of coding paradigm, a standardized platform would result and likewise adding new functionality would be possible in a more simplified matter, with the added benefit of reconfiguration of the model being done through one set of fields for selecting the appropriate submodules.

CONCLUSION

The primary goal of this thesis was to address the need for reducing the calculation time of the SOFC model used in the Hyper facility in NETL, due to the cyber-physical fuel cell only being capable of responding to stimuli as quickly as its computational components can calculate said response. This was achieved through the reduction of the model computational time by an order of magnitude thus allowing the cyber-physical solid oxide fuel cell to respond at a higher sampling rate, and subsequently resolve transient behavior more accurately and at a higher temporal resolution. To this aim, several aspects of the numerical model were investigated, with the electrochemical model and the thermal model being the key areas of examination. It was determined that a majority of the calculation time was spent in the electrochemical routine, specifically resolving the cell voltage and local current density.

An investigation into the formulation of the electrochemical solver discovered two possible outlets of optimization, first through the redefinition of the convergence parameters to ensure a relative tolerance of 1e-3, and secondly to the numerical recipes used for resolving the cell voltage and local current density. Due to the nature of the underlying governing electrochemical equations, a highly iterative dual rootfinding scheme was necessary for concurrently resolving the voltage and current density throughout the cell. The rootfinders used in the voltage and current density algorithms were replaced with higher order techniques, capable of better convergence performance, to create a set of new electrochemical algorithms for testing. This coupled with the new definitions for the convergence parameters, lead to creating several new accelerated algorithms. These

accelerated algorithms specifically replaced the bisection and false position based approaches, with secant based numerical recipes instead. Likewise, the accelerated algorithms required the implementation of a current density extrapolation scheme, adding an adaptive functionality to the current density algorithm. This extrapolation technique allows the current density algorithm to terminate early while the relative error is high, saving computational power by resolving the full current density profile only when necessary. Upon testing on the Hyper platform, both offline and fully coupled to the turbomachinery, a reduction in calculation time from 80 milliseconds to below 5 milliseconds was achieved using the accelerated electrochemical algorithms, indicating an order of magnitude reduction in calculation time.

Furthermore, an investigation into the thermal algorithm used for calculating the solid and gaseous thermal fields was performed. The original thermal algorithm using the Crank-Nicolson method was replaced with a different time-stepping scheme utilizing a three-point backwards time discretization technique and a fully implicit formulation with the aim to improve transient response and improving stability during severe transient events. The results from this investigation showed similar behavior as to the Crank-Nicolson scheme, with nearly identical computational performance on the Hyper facility, showing that a higher order fully implicit scheme can be implemented on Hyper if need be with sub 5 millisecond calculation time.

Upon the conclusion of the original investigation, the original dSPACE hardware powering the simulation platform of Hyper was replaced with a newer model. With that the calculation time of all models was further reduced substantially, with the fastest accelerated models dropping in calculation time from approximately 5 milliseconds to a

maximum of approximately 1.5 milliseconds. The results still indicated that the accelerated methods utilizing higher order rootfinding schemes were necessary to consistently achieve sub 5 millisecond calculation times, despite the new dSPACE hardware.

Lastly, with the fastest SOFC model being able to consistently calculate below 2 milliseconds using the accelerated methods, a preliminary investigation into the effects of model resolution was initiated. This revealed that significant sensitivity to the mesh density was present in the model for extreme cases. Proposed solutions for investigation in future scenarios include a new discretization of the oxidant phase governing PDE for better reproduction of the effects of convection in the gaseous channel, and likewise variable discretization of the domain in order to better resolve areas in the mesh subject to large gradients in temperature.

Overall, the study shows that the SOFC model used by Hyper was able to be successfully optimized to operate at a sub 5 millisecond calculation time, i.e. an order of magnitude reduction in calculation time. The ultimate result of this means the cyber-physical solid oxide fuel cell can resolve all transient features of interest at higher temporal resolution and faster sampling rates, due to the reduction in time necessary for calculation. Likewise, the additional time available for computation means that additional physics can be implemented into Hyper such as degradation effects, more complex mass transport, or higher resolution if need be. This result is key for maintaining the flexibility of the Hyper platform when testing a plethora of SOFC/gas turbine hybrids and their behavior under a vast range of plausible test cases in addition to further ranging studies such as integration into micro grids, smart grids featuring renewables with rapidly varying power generation and demand, or whatever else the future of the power grid may bring.

APPENDIX A. SOFC MODEL OPERATIONAL PARAMETERS

Table A.1: Model Constants and Parameters [17]

Diffusion Polarization	
Pore diameter, d_{pore} (anode and cathode)	1×10^{-6} m
Porosity, ε (anode and cathode)	0.5
Tortuosity, τ (anode and cathode)	3
Activation Polarization	
Pre-exponential factors	
Anode, γ_{an}	5.5×10^{8} A/m^2
Cathode, γ_{ca}	7×10^{8} A/m^2
Activation Energies	
Anode, E_{an}	50 kJ/mol
Cathode, E_{ca}	100 kJ/mol
Transfer coefficient, α	0.5
Ohmic Polarization	
Temp. dependent resistivities, ρ	
Anode	$\left[\frac{95\times10^{6}}{T_{PEN}}\exp\left(-\frac{1150}{T_{PEN}}\right)\right]^{-1}\ \Omega-m$
Cathode	$\left[\frac{42\times10^{6}}{T_{PEN}}\exp\left(-\frac{1200}{T_{PEN}}\right)\right]^{-1}\ \Omega-m$
Electrolyte	$\left[3.34\times10^{4}\exp\left(-\frac{10.300}{T_{PEN}}\right)\right]^{-1}\ \Omega-m$
Direct Internal Reformation	
Pre-exponential factor, γ_{sr}	4274 mol/s-m^2-bar
α Coefficient	1
β Coefficient	0
Activation Energies, E_{an}	82,000 J/mol
Diffusion Coefficient Correction	
C	4.88×10^{-4}
N	2
i_{ref}	1 A/m^2
Diffusion Volumes	
H_2	6.12
H_2O	13.1
O_2	16.3
N_2	18.5

Table A.2: Nominal Initialization Conditions for Testing

Fuel Properties	
Composition: Syngas	
CH_4	1e-6 %
H_2	29.1 %
CO	28.6 %
CO_2	12.0 %
H_2O	27.1 %
Flow Rate	0.103 kg/s
Pressure	240 kPa
Temperature	800 °C
Air Properties	
Composition: Air	
N_2	79 %
O_2	21 %
Flow Rate	1.02 kg/s
Pressure	240 kPa
Temperature	700 °C

REFERENCES

[1] Tucker, D., Pezzini, P., and Bryden, K. M., 2018, "Cyber-Physical Systems: A New Paradigm for Energy Technology Development," ASME 2018 Power Conference, 1.

[2] Tucker, D., Harun, N. F., Zaccaria, V., Bryden, K. M., and Haynes, C., 2017, "Real-Time Fuel Cell Model Development Challenges for Cyber-Physical Systems in Hybrid Power Applications."

[3] Hebin, W., and Rui, B., 2018, "Design of Hardware-In-Loop Simulation Platform for Air Suspension Systems," IFAC-PapersOnLine, 51(31), pp. 142-145.

[4] Kendall, I. R., and Jones, R. P., 1999, "An investigation into the use of hardware-in-the-loop simulation testing for automotive electronic control systems," Control Engineering Practice, 7(11), pp. 1343-1356.

[5] Lachaize, J., Lamamy, R., and Verdier, D., 2017, "Model of a Hybrid Electrical System for Software and System V & V on Hardware In the Loop Test Bench," IFAC-PapersOnLine, 50(1), pp. 7863-7868.

[6] Guo, L., Li, J., and Fu, Z., 2019, "Lithium-Ion Battery SOC Estimation and Hardware-in-the-Loop Simulation Based on EKF," Energy Procedia, 158, pp. 2599-2604.

[7] Arias, J. E., Haynes, C. L., and Giorges, A. G., 2018, "Resolving the Electrochemical Equations of a Solid Oxide Fuel Cell for Use in Transient Simulation and Integration Into Cyber-Physical Systems."

[8] Smith, T. P., Haynes, C. L., Wepfer, W. J., Tucker, D., and Liese, E. A., 2006, "Hardware-Based Simulation of a Fuel Cell Turbine Hybrid Response to Imposed Fuel Cell Load Transients," (47640), pp. 319-328.

[9] Williams, M. C., Strakey, J., and Sudoval, W., 2006, "U.S. DOE fossil energy fuel cells program," Journal of Power Sources, 159(2), pp. 1241-1247.

[10] Rao, A. D., and Samuelsen, G. S., 2002, "A Thermodynamic Analysis of Tubular Solid Oxide Fuel Cell Based Hybrid Systems," Journal of Engineering for Gas Turbines and Power, 125(1), pp. 59-66.

[11] Smith, T. P., 2007, "Hardware Simulation of Fuel Cell/Gas Turbine Hybrids," Georgia Institute of Technology, Atlanta.

[12] Hughes, D., Wepfer, W. J., Davies, K., Ford, J. C., Haynes, C., and Tucker, D., 2011, "A Real-Time Spatial SOFC Model for Hardware-Based Simulation of Hybrid Systems," (54693), pp. 409-428.

[13] Kakaç, S., Pramuanjaroenkij, A., and Zhou, X. Y., 2007, "A review of numerical modeling of solid oxide fuel cells," International Journal of Hydrogen Energy, 32(7), pp. 761-786.

[14] Pramuanjaroenkij, A., Kakaç, S., and Zhou, X. Y., "Mathematical Analysis of Planar Solid Oxide Fuel Cells," Springer Netherlands, pp. 359-390.

[15] Wang, K., Hissel, D., Péra, M. C., Steiner, N., Marra, D., Sorrentino, M., Pianese, C., Monteverde, M., Cardone, P., and Saarinen, J., 2011, "A Review on solid oxide fuel cell models," International Journal of Hydrogen Energy, 36(12), pp. 7212-7228.

[16] Liese, E. A., Gemmen, R. S., Smith, T. P., and Haynes, C. L., 2006, "A Dynamic Bulk SOFC Model Used in a Hybrid Turbine Controls Test Facility," (42398), pp. 117-126.

[17] Hughes, D., 2011, "A Hardware-Based Transient Characterization of Electrochemical Start-Up in an SOFC/Gas Turbine Hybrid Environment Using a 1-D Real Time SOFC Model ", Georgia Institute of Technology, Atlanta.

[18] Gemmen, R. S., Liese, E., Rivera, J. G., Jabbari, F., and Brouwer, J., 2000, "Development of Dynamic Modeling Tools for Solid Oxide and Molten Carbonate Hybrid Fuel Cell Gas Turbine Systems."

[19] Li, P.-W., Kotwal, A., Sepulveda, J. L., Loutfy, R. O., and Chang, S., 2009, "An Easy-to-Approach Comprehensive Model and Computation for SOFC Performance and Design Optimization," Inaugural US-EU-China Thermophysics Conference-Renewable Energy 2009 (UECTC 2009 Proceedings), Y. Tao, and C. Ma, eds., ASME, New York, NY.

[20] Campanari, S., and Iora, P., 2005, "Comparison of Finite Volume SOFC Models for the Simulation of a Planar Cell Geometry," Fuel Cells, 5(1), pp. 34-51.

[21] Noren, D. A., and Hoffman, M. A., 2005, "Clarifying the Butler–Volmer equation and related approximations for calculating activation losses in solid oxide fuel cell models," Journal of Power Sources, 152, pp. 175-181.

[22] Li, M., Brouwer, J., Powers, J. D., and Samuelsen, G. S., 2009, "A Finite Volume SOFC Model for Coal-Based Integrated Gasification Fuel Cell System Analysis," (48814), pp. 677-686.

[23] Cayan, F. N., Pakalapati, S. R., Elizalde-Blancas, F., and Celik, I., 2008, "On Modeling Multi-Component Diffusion Inside the Porous Anode of Solid Oxide Fuel Cells Using Fick's Model," (43181), pp. 377-386.

[24] Süli, E., and Mayers, D. F., 2003, An Introduction to Numerical Analysis, Cambridge University Press, Cambridge.

[25] Thomas, L. H., 1949, "Elliptic Problems in Linear Differential Equations over a Network," Watson Science Computer Laboratory Report, Columbia University, New York, NY.

[26] Burden, R. L., and Faires, J. D., 2011, Numerical analysis, Brooks/Cole, Cengage Learning, Boston, MA.

[27] Patankar, S. V., 1980, Numerical heat transfer and fluid flow, Hemisphere Pub. Corp., McGraw-Hill, Washington, New York.

[28] Ravnik, J., and Škerget, L., 2011, BEM simulation of transient fluid flow phenomena.

[29] Langtangen, H. P., 2017, Finite difference computing with PDEs : a modern software approach, Springer International Pub., New York, NY.